

applications & TOOLS

**PC-basierte Automatisierung mit
SIMATIC WinAC**

Positionieren mit WinAC RTX, Easy Motion Control
und MICROMASTER 4xx



SIEMENS

Präambel

Vorwort

Das SIMATIC Windows Automation Center (WinAC) bietet Ihnen eine offene, vielseitige und robuste Basis, um Ihre Automatisierungslösung PC-basiert zu verwirklichen.

Durch volle Kompatibilität zu SIMATIC S7, kombiniert mit Schnittstellen zur (offenen) PC-Welt, können Sie Ihre klassischen Automatisierungsaufgaben mit den Möglichkeiten der PC-Welt verbinden. Der Einsatz von SIMATIC Industrie PCs bietet eine leistungsfähige und robuste Plattform für Ihre Automatisierungslösung mit WinAC.

Zum schnellen Einstieg in die PC-basierte Automatisierung mit SIMATIC WinAC wurden acht Beispiele entwickelt. Sie bestehen aus Beispielcode und einer ausführlichen Dokumentation. Anhand dieser Beispiele können Sie als Anwender sich aufgabenbezogen in die einzelnen Themen einarbeiten.

Einteilung der Beispiele

Die in den Beispielen behandelten Automatisierungsaufgaben orientieren sich dabei an typischen Aufgabenbereichen der Automatisierungstechnik:

- Steuern
- Kommunikation
- Visualisierung
- Technologie

Positionieren mit WinAC RTX

Die einzelnen Beispiele

Um die PC-basierte Automatisierung optimal einsetzen zu können, haben wir für die vier typischen Aufgabenbereiche (Steuern, Kommunikation, Visualisierung, Technologie) jeweils ein Beispiel aus der „klassischen“ SPS-Welt und jeweils eines aus der „offenen“ PC-Welt entwickelt.

Die folgende Abbildung zeigt alle acht Beispiele mit ihrer Zuordnung zu den jeweiligen Aufgabenbereichen. Das vorliegende Beispiel zum Thema „Positionieren mit WinAC RTX“ ist mit einem roten Rand markiert.

Abbildung 1-1

Beispiele zum Thema PC-basierte Automatisierung				
	Steuern	Visualisieren	Technologie	Kommunikation
Beispiele aus der klassischen Welt	Automatisieren mit WinAC RTX • Einrichten der PC Plattform • Prozessanschluss für WinAC RTX • Konfiguration von WinAC ①	Zusammenspiel von WinCC und WinAC • Tuning für WinAC RTX • Anbinden von WinCC an WinAC RTX • Visualisieren mit WinCC ③	Positionieren mit WinAC RTX • Positionieren mit Easy Motion Control • Anbindung eines MICROMASTER 440 ⑤	S7-Kommunikation mit WinAC RTX • S7-Kommunikation mit BSEND/BRCV • über Industrial Ethernet • Kommunikation: WinAC ↔ S7-300 ⑦
Beispiele aus der offenen Welt	Datenaustausch über OPC XML • dynamische Rezepturdaten-Steuerung • Datenaustausch zur Datenbank über OPC XML • Datenbankschnittstelle ADO.NET ②	Visualisieren mit Visual Basic.NET + OPC XML • Zugriff auf OPC XML über VB.NET • Webserver mit IIS einrichten • Lokale und remote Visualisierung ④	SIMATIC Vision Sensoren und WinAC ODK • Konfiguration des Vision Sensors • Anwendung der CCX Schnittstelle von WinAC ODK • Kommunikation Vision Sensor ↔ ODK ⑥	E-Mail versenden mit WinAC RTX • Anwendung der SMX Schnittstelle von WinAC ODK • Versenden von E-Mail über die MAPI Schnittstelle ⑧

Positionieren mit WinAC RTX

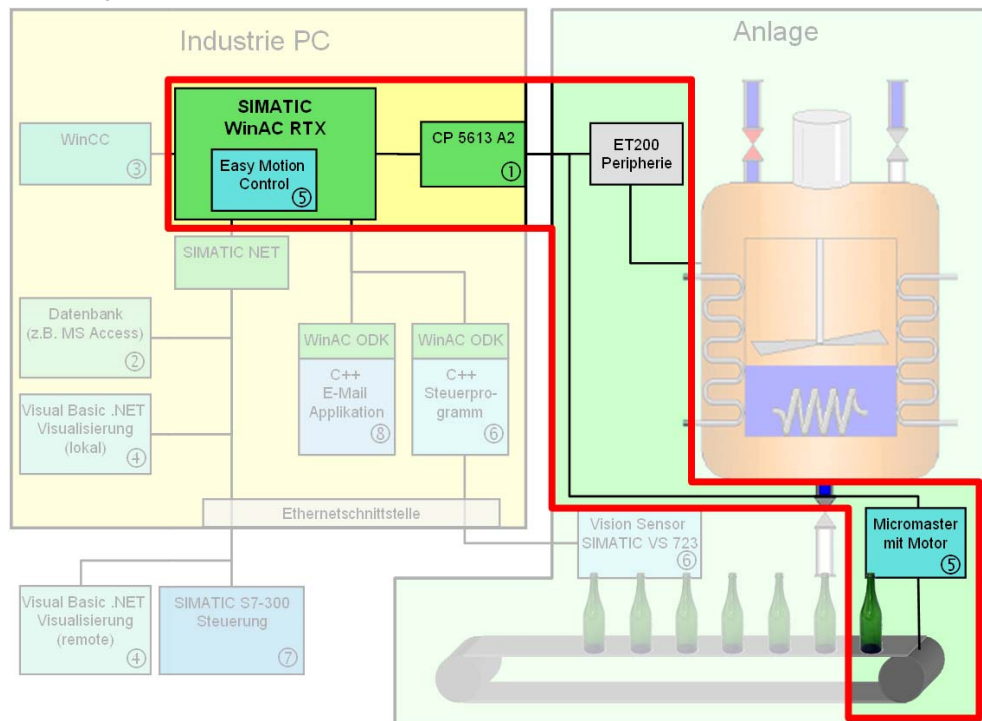
Grundlage der Beispiele

Als gemeinsamer Aufhänger ist allen Beispielen ein fiktiver „Mischprozess“ zu Grunde gelegt. Anhand dieses „Mischprozesses“ werden die unterschiedlichen Aufgabenstellungen und Automatisierungskomponenten aus dem Produktspektrum der PC-basierten Automatisierung appliziert.

Anlagenbild

Folgendes Bild zeigt das Anlagenbild des „Mischprozesses“. Der rote Rahmen zeigt Ihnen, welche Komponenten in dem vorliegenden Beispiel behandelt werden.

Abbildung 1-2



Gewährleistung, Haftung und Support

Für die in diesem Dokument enthaltenen Informationen übernehmen wir keine Gewähr.

Unsere Haftung, gleich aus welchem Rechtsgrund, für durch die Verwendung der in diesem Dokument beschriebenen Beispiele, Hinweise, Programme, Projektierungs- und Leistungsdaten usw. verursachte Schäden ist ausgeschlossen, soweit nicht z.B. nach dem Produkthaftungsgesetz in Fällen des Vorsatzes, der grober Fahrlässigkeit, wegen der Verletzung des Lebens, des Körpers oder der Gesundheit, wegen einer Übernahme der Garantie für die Beschaffenheit einer Sache, wegen des arglistigen Verschweigens eines Mangels oder wegen Verletzung wesentlicher Vertragspflichten zwingend gehaftet wird. Der Schadensersatz wegen Verletzung wesentlicher Vertragspflichten ist jedoch auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht Vorsatz oder grobe Fahrlässigkeit vorliegt oder wegen der Verletzung des Lebens, des Körpers oder der Gesundheit zwingend gehaftet wird. Eine Änderung der Beweislast zu Ihrem Nachteil ist hiermit nicht verbunden.

Die Applikationsbeispiele sind unverbindlich und erheben keinen Anspruch auf Vollständigkeit hinsichtlich Konfiguration und Ausstattung sowie jeglicher Eventualitäten. Sie stellen keine kundenspezifische Lösungen dar, sondern sollen lediglich Hilfestellung bieten bei typischen Aufgabenstellungen. Sie sind für den sachgemäßen Betrieb der beschriebenen Produkte selbst verantwortlich. Diese Applikationsbeispiele entheben Sie nicht der Verpflichtung zu sicherem Umgang bei Anwendung, Installation, Betrieb und Wartung. Durch Nutzung dieses Applikationsbeispiels erkennen Sie an, dass Siemens über die oben beschriebene Haftungsregelung hinaus nicht für etwaige Schäden haftbar gemacht werden kann. Wir behalten uns das Recht vor, Änderungen an diesem Applikationsbeispiel jederzeit ohne Ankündigung durchzuführen. Bei Abweichungen zwischen den Vorschlägen in diesem Applikationsbeispiel und anderen Siemens Publikationen, wie z.B. Katalogen, hat der Inhalt der anderen Dokumentation Vorrang.

Copyright© 2004 Siemens A&D. Weitergabe oder Vervielfältigung dieser Applikationsbeispiele oder Auszüge daraus sind nicht gestattet, soweit nicht ausdrücklich von Siemens A&D zugestanden.

Bei Fragen zu diesem Beitrag wenden Sie sich bitte über folgende E-Mail-Adresse an uns:

csweb@ad.siemens.de

Positionieren mit WinAC RTX

Inhaltsverzeichnis

1	Aufgabenstellung	7
2	Aufbau der Automatisierungslösung	9
2.1	Darstellung der beteiligten Komponenten.....	9
2.2	Erforderliche Komponenten	10
2.3	Leistungseckdaten	13
3	Funktionsmechanismen und Programmstrukturen	14
3.1	WinAC RTX Grundlagen.....	14
3.2	Weitere Grundlagen.....	14
3.3	Projektierung der Hardware	17
3.4	Kernelemente dieser Lösung	17
3.5	Logische Kernstruktur im Beispiel	23
3.6	Programmstrukturen	24
3.7	Programmablauf	25
4	Installation der Hard- und Software	29
4.1	Vorbereitende Installation	29
4.2	Hardwareaufbau	31
4.2.1	Industrie PC / WinAC (Station 1)	31
4.2.2	ET200S Station (Station 2)	31
4.2.3	MICROMASTER 440 (Station 3) und Motor	32
4.3	Installation der WinAC RTX Software	40
4.4	Benutzung von WinAC RTX	40
4.5	Konfiguration der Automatisierungsstationen	40
5	Bedienen und Beobachten	44
6	Literaturangaben	50

1 Aufgabenstellung

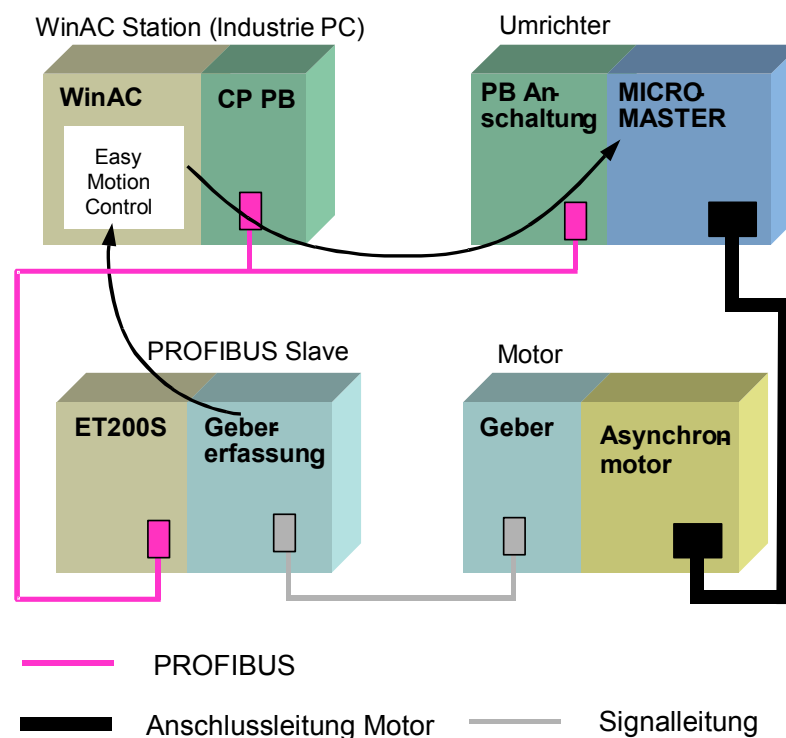
Aufgabenstellung/ Übersicht

Die Lageregelung wird in der WinAC mit Hilfe der Easy Motion Control Bausteine realisiert. Die für den Umrichter nötigen Positioniersignale werden über den PROFIBUS gesendet. Am Motor befindet sich ein inkrementeller Drehzahlgeber, mit dessen Hilfe eine Erfassung der momentanen Motorgeschwindigkeit und Achsposition erfolgt. Die Gebererfassung befindet sich auf der dezentralen Peripheriebaugruppe ET 200S, die über den PROFIBUS mit der WinAC kommuniziert und so den Istwert der Lage zur Regelung zurück liefert.

Prinzipbild der Automatisierungslösung

Das folgende Bild zeigt prinzipiell das Zusammenspiel der Automatisierungskomponenten, die in diesem Beispiel behandelt werden.

Abbildung 1-1



Positionieren mit WinAC RTX

Anforderungen an die Lösung

Die Lösung muss folgende Punkte erfüllen:

- Ein STEP7 Programm in der S7-CPU (WinAC), welches die Easy Motion Control Bausteine enthält, steuert den MICROMASTER an. Die Bedienung erfolgt über eine Variablen-tabelle.
- Simuliert wird der Flaschentransport einer Abfüllanlage für Erfrischungsgetränke (die Flaschen werden unter der Abfülleinrichtung positioniert). Folgender prinzipieller Verfahrensablauf kommt dabei zum Einsatz:
 - Band steht zunächst
 - über eine Startrampe fährt das Band an
 - Band fährt eine gewisse Strecke (parametrierbar über Variablen-tabelle)
 - Band bremst ab (Rampe)
 - Band steht für eine gewisse Zeit (Abfüllung, parametrierbar über Variablen-tabelle)
 - nach Ablauf der Zeit wird die Prozedur fortlaufend wiederholt
- Funktionalität auf große Mengengerüste (mehrere Achsen) anwendbar

Das Mengengerüst des Beispiels

In der folgenden Tabelle werden die im Projekt auftretenden Datenmengen grob aufgeführt. Betrachtet wird jeweils nur eine Achse.

Tabelle 1-1

Kriterium	Leitungseckdaten
Datenmenge (Easy Motion Control und Anwenderprogramm)	Ca. 1kByte
Ladespeicherbedarf	34 kByte
Arbeitsspeicherbedarf	24 kByte
Übertragungsraten (PROFIBUS)	1,5 Mbit /sec

Kundennutzen

Im Vergleich zu klassischen Positionieraufgaben, die ohne PC durchgeführt werden, profitiert der Kunde bei dieser Lösung in folgender Weise:

- Schnelle Bearbeitung der Positionieraufgaben auch bei mehreren Achsen
- Gleichzeitige Visualisierung der Steuerung (Steuerung und Visualisierung in einem Gerät).
- Im Vergleich zu Hardwaresteuerungen sind ca. fünf mal mehr Achsen verwendbar, sofern die Kommunikation der Peripherie dies zulässt.

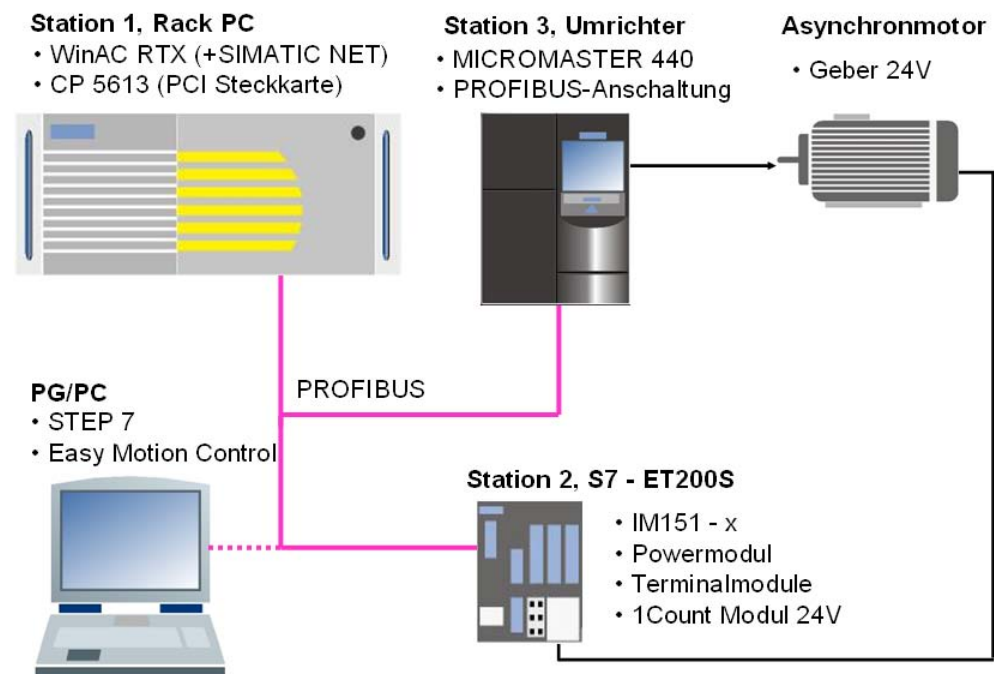
2 Aufbau der Automatisierungslösung

In diesem Kapitel wird der hardwaremäßige Aufbau der Automatisierungslösung gezeigt. Im Kapitel 2.2 Erforderliche Komponenten werden tabellarisch die jeweils benötigten Hard- und Softwarekomponenten aufgelistet.

2.1 Darstellung der beteiligten Komponenten

Das folgende Übersichtsbild zeigt den HW-Aufbau der Beispielapplikation und der darauf befindlichen Standard- und Anwendersoftwarekomponenten.

Abbildung 2-1



2.2 Erforderliche Komponenten

Nachfolgende Tabellen enthalten die Hard- und Softwarekomponenten für die jeweiligen Stationen.

Hardwarekomponenten Station 1 (WinAC Station)

Als Industrie PC kann alternativ auch ein beliebiger anderer (aktueller) Industrie PC von Siemens Verwendung finden. Die unten aufgeführten Komponenten können Sie direkt in der Siemens A&D Mall unter <https://mall.automation.siemens.com> bestellen.

Die angegebenen FAQs finden Sie auf der A&D Support Homepage unter <http://support.automation.siemens.com> (Geben Sie dort die FAQ ID in das Suchfeld ein).

Tabelle 2-1

Komponente	Anz.	MLFB/Bestellnummer	Hinweis
Industrie PC SIMATIC Rack PC IL 40 S	1	6AG4011-0CA21-0JX0	Konfigurator: siehe FAQ ID 17128155
Kommunikationsprozessor CP 5613 A2 für PROFI-BUS, PCI-KARTE	1	6GK1 561-3AA00	

Hardwarekomponenten Station 2 (S7-ET200S)

Die ET200S Station kann von beliebigem Typ sein, da das Countermodul von jedem Interfacemodul unterstützt wird.

Tabelle 2-2

Komponente	Anz.	MLFB/Bestellnummer	Hinweis
Stromversorgung PS 307 2A	1	6ES7 307-1BA00-0AA0	
Adapter für Normprofil-schiene	1	6ES7 390-6BA00-0AA0	Zum Montieren auf die Hutschiene.
Hutschiene 500mm	1	8GR4 926	
IM 151-1	1	6ES7 151-1AA01-0AB0	Auf Hutschiene zu montieren
Terminalmodul (für Powermodul) TM-P15S23-A1	1	6ES7 193-4CC20-0AA0	Auf Hutschiene zu montieren
Terminalmodul (für Countermodul) TM-E15S24-01	1	6ES7 193-4CB20-0AA0	Auf Hutschiene zu montieren
Powermodul PM-E DC24V	1	6ES7 138-4CA00-0AA0	
Countermodul 1Count 24V/1024 Inkremente	1	6ES7 138-4DA03-0AB0	

Positionieren mit WinAC RTX

Hardwarekomponenten Station 3 (MICROMASTER 440)

Tabelle 2-3

Komponente	Anz.	MLFB/Bestellnummer	Hinweis
MICROMASTER 440	1	6SE6440-2UC12-5AA1	
PC_Umrichter-Verbindungssatz	1	6SE6400-1PC00-0AA0	Kann entfallen, falls das PG über einen PROFIBUS Anschluss verfügt, oder die CPU routingfähig ist. Voraussetzung ist eine serielle Schnittstelle am PC.
MICROMASTER 4 PROFIBUS MODUL	1	6SE6400-1PB00-0AA0	

Hardwarekomponenten für Motor

Tabelle 2-4

Komponente	Anz.	MLFB/Bestellnummer	Hinweis
Norm – Asynchronmotor 120W	1	1LA7060-4AB10-Z Optionen A23	Die Option A23 ermöglicht eine Motortemperaturauswertung. Detaillierte Informationen finden Sie in der Betriebsanleitung des MICRO-MASTERS im Kapitel 3.21
Inkrementalgeber 1024 Inkremente / 24V	1	1XP8001-1	Wird eingeschraubt. Alternativ dazu kann auch jeder andere, für den verwendeten Motor zulässige Geber mit 24V Spannung, 1024 Inkrementen und 2 Spuren (Richtungserkennung) verwendet werden

Positionieren mit WinAC RTX

Hardwarekomponenten für das PG/PC

Tabelle 2-5

Komponente	Anz.	MLFB/Bestellnummer	Hinweis
Programmiergerät Power PG	1	6ES7 750-2CA52-4FB4	Konfigurator: siehe FAQ ID 17128155 ; CP 5611 integriert

Softwarekomponenten auf der SIMATIC PC-Station (Station 1)

Tabelle 2-6

Komponente	Anz.	MLFB/Bestellnummer	Hinweis
SIMATIC WinAC RTX V4.1	1	6ES7 671-0RC04-0YA0	SIMATIC NET CD ist im WinAC Paket enthalten.
Easy Motion Control V2.0	1	6ES7 864-0AC01-0YX0	

Softwarekomponenten auf dem PG/PC

Tabelle 2-7

Komponente	Anz.	MLFB/Bestellnummer	Hinweis
STEP 7 V5.3	1	6ES7 810-4CC07-0YA5	
STARTER V3.0	1		Downloadbar unter der Beitrags ID 13336809
ProTool / Pro V6.0+SP2	1	6AV6 582-2BX06-0CX0	Optional
Drive ES	1	6SW1 700-0JA00-0AA0	Optional

Alternativ zur angegebenen Konfiguration kann zum Bedienen und Beobachten anstelle einer Variablen-tabelle auch ProTool / Pro verwendet werden. Mit Drive ES kann eine einfache Integration der Motorparametrierung in STEP 7 erfolgen.

Beispielprojekt

Die vorliegende Beispielapplikation besteht aus folgenden Komponenten.

Weitere Informationen zur Inbetriebnahme von Hard- und Software entnehmen Sie bitte dem Kapitel 4 Installation der Hard- und Software.

Tabelle 2-8

Komponente	Hinweis
21004767_WinAC_TK_CODE_v10.zip	Diese gepackte Datei enthält den vollständigen Code und die Konfigurationen für diese Applikation
21004767_WinAC_TK_DOKU_v10_d.pdf	Dieses Dokument

2.3 Leistungseckdaten

Nachfolgende Tabelle informiert über die Eckdaten der Systemsoftware und Projektierung. Sie erhalten damit einen Überblick über die Leistungsfähigkeit dieser Beispielapplikation und ihrer Komponenten.

Tabelle 2-9

Merkmal	WinAC RTX
Ladespeicherbedarf für EMC Bausteine	Ca. 25 kByte
Arbeitsspeicherbedarf für EMC Bausteine	Ca. 21 kByte
Lokaldaten der EMC Bausteine	Ca. 324 Byte
Ladespeicherbedarf für Applikation	Ca. 2 Kbyte
Arbeitsspeicherbedarf für Applikation	Ca. 1,5 kByte
Gesamtzeit zur Ausführung der EMC Bausteine im Beispielprogramm (ohne Applikationsbausteine, mit einer Achse) gemessen mit WinAC RTX V4.1 auf Athlon mit 1333 MHz	162 µs (typischer Wert)

Hinweis

Die Leistungsfähigkeit der gewünschten Positionierung richtet sich nach dem im Industrie PC eingesetzten Prozessor, der verwendeten Peripherie und der zu Grunde liegenden Kommunikationsgeschwindigkeit.

3 Funktionsmechanismen und Programmstrukturen

In diesem Kapitel werden kurz die Grundlagen sowie die Benutzung von WinAC RTX erklärt. Zusätzlich wird auf die grundsätzliche Arbeitsweise von Easy Motion Control eingegangen.

Weiterhin beschreibt es die in der vorliegenden Beispielapplikation benutzten Easy Motion Control Bausteine und geht auf deren spezielle Eigenschaften ein.

Behandelte Themen

Kap.	Titel	Seite
3.1	WinAC RTX Grundlagen	14
3.2	Weitere Grundlagen	14
3.3	Projektierung der Hardware	17
3.4	Kernelemente dieser Lösung	17
3.5	Logische Kernstruktur im Beispiel	23
3.6	Programmstrukturen	24
3.7	Programmablauf	25

3.1 WinAC RTX Grundlagen

Die Grundlagen zur WinAC RTX entnehmen Sie bitte dem Kapitel 3 der Applikation „Grundlagen für Lösung von Automatisierungsaufgaben auf Basis von WinAC RTX“ (Beitrags-ID: [21004765](#)).

3.2 Weitere Grundlagen

Frequenzumrichter allgemein

Ein Frequenzumrichter ermöglicht auf Basis einer speziellen Leistungselektronik, verbunden mit einem Mikroprozessor, eine Regelung der Drehzahl des Motors über die Frequenz (f) der angelegten Spannung (U) bzw. des Stromes. Dabei kann das Verhältnis U/f (proportional n) konstant gehalten oder aber ein errechneter Strom eingeprägt werden, wodurch eine sehr genaue elektronische Regelung bzw. Steuerung der Drehzahl erreicht wird. Mit Hilfe dieser Technologie kann ein Asynchronmotor nun auch für Positionieraufgaben eingesetzt werden.

Im Folgenden wird der Frequenzumrichter mit Umrichter bezeichnet.

Frequenzumrichter MICROMASTER 440

Der MICROMASTER 440 ist ein Umrichter für Drehstrommotoren mit zusätzlicher Drehzahl- und Drehmomentregelung. Ein Mikroprozessor regelt oder steuert die verschiedenen Funktionen des Motors. Der Umrichter unterstützt verschiedene Verfahren zur Ansteuerung des Motors. Darüber hinaus verfügt er zusätzlich über automatische Motorenidentifikationsmechanismen, verschiedenste logische Verkettungen, Schutzfunktionen und vieles mehr. Zur Handhabung dieser Funktionsvielfalt stehen über 3000 Parameter zur Verfügung.

Mit Hilfe von aufsteckbaren Bedienpanels, PC Verbindungskits oder Bus-optionsmodulen kann der Umrichter parametrierbar, getestet und bedient werden.

Für detaillierte Informationen schlagen Sie bitte in der Betriebsanleitung des MICROMASTERS (Kapitel 8) nach.

Drehzahlregelung / Geberlose Vektorregelung (SLVC)

In modernen Umrichtern sind Drehzahlregler meist gleich mit integriert. Beim MICROMASTER 440 wird zur Drehzahlregelung eine so genannte feldorientierte Vektorregelung (kurz: Vektorregelung) verwendet. Das Prinzip der Vektorregelung beruht auf der Einprägung eines Motorstromes in Abhängigkeit vom Motorfluss in der Weise, dass sich anschließend das gewünschte Moment am Motor einstellt. Die Drehzahl würde trotz einer erhöhten Last an der Achse schließlich auf den gewünschten Wert zurückgeregelt werden. Für die Bestimmung des einzuprägenden Stromes verwendet der Umrichter verschiedene mathematische Algorithmen sowie das mit der Motoridentifikation ermittelte theoretische Motormodell.

Für die Verwendung dieser Drehzahlregelung ist im MICROMASTER unbedingt eine Motoridentifikation durchzuführen.

Easy Motion Control im Überblick

Easy Motion Control ist ein Softwarepaket zur einfachen Positionierung von Antrieben. Die Positionierungen werden mit Hilfe eines Softwarelagereglers bewerkstelligt, welcher eine genaue Positionierung (abhängig von der eingesetzten Hardware) ermöglicht. Die Regelung der Lage wird dabei ausschließlich von der Software abgewickelt. Zur Anbindung der Peripherie ist keine spezielle Hardware nötig. Der MICROMASTER 440 könnte auch durch konventionelle Leistungsteile ersetzt werden, bietet aber den Vorteil, zusätzlich noch eine Drehzahl- / Drehmomentregelung und verschiedene Schutzfunktionen bereitzustellen.

Positionieren mit WinAC RTX

Funktionen von Easy Motion Control

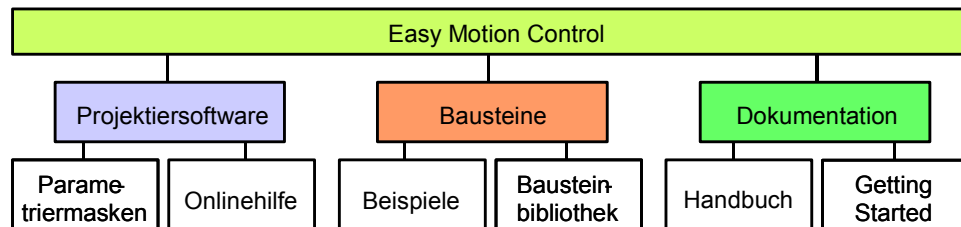
Easy Motion Control stellt grundsätzlich folgende Funktionen zur Verfügung:

- Tippen
- Referenzpunktfahren
- Positionieren absolut/relativ
- Elektronisches Getriebe
- Bezugspunkt setzen
- Alle gängigen Überwachungsfunktionen für Positionierung
- Geschwindigkeits-Override
- Lageregelung
- Simulation
- Vorkonfektionierte Ein- und Ausgangstreiber für Geber- bzw. Analogausgabebaugruppen

Komponenten von Easy Motion Control

Mit Easy Motion Control werden folgende Komponenten ausgeliefert.

Abbildung 3-1



Positionieren mit WinAC RTX

3.3 Projektierung der Hardware

Projektierung der Steuerung

Die WinAC wird mit einem PROFIBUS Kommunikations-Prozessor projektiert, der die Kommunikation zum MICROMASTER und der ET200S Station ermöglicht.

Projektierung des Umrichters

Der MICROMASTER wird im STEP 7 als DP Slave projektiert und mit der am MICROMASTER eingestellten Adresse parametrisiert. (Die Adresse wird auf der PROFIBUS Anschaltbaugruppe mit Hilfe von DIL Schaltern eingestellt.)

Projektierung der ET200S Station

Die ET200S Station wird als Slave projektiert und am Interfacemodul wird die in der HW-Konfig projektierte Adresse eingestellt.

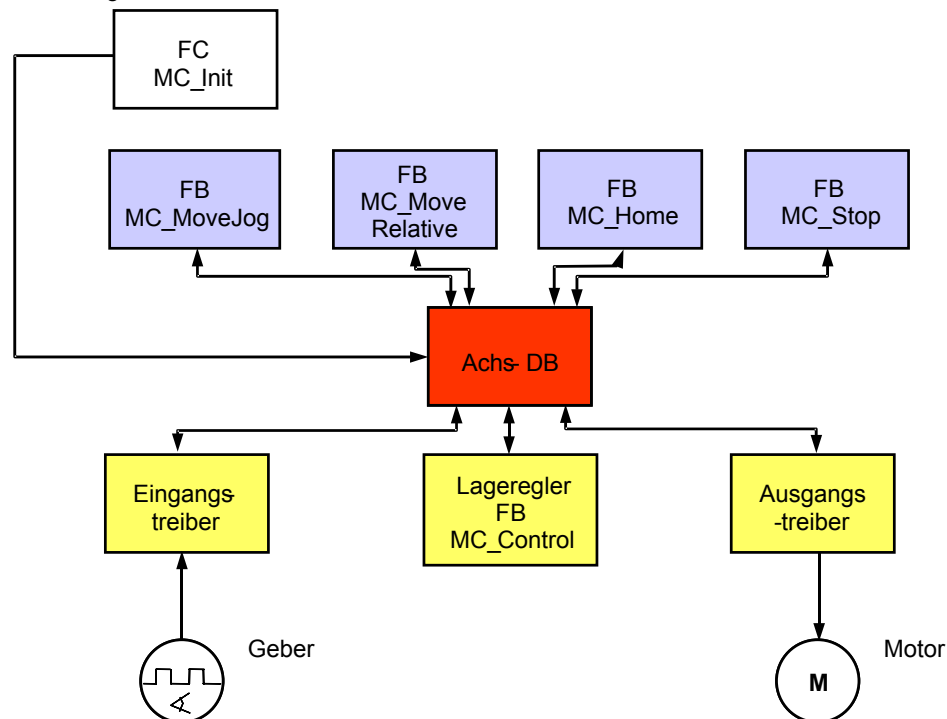
3.4 Kernelemente dieser Lösung

Wichtige Easy Motion Control Bausteine

Die grundlegenden Bausteine und der prinzipielle Datenfluss von Easy Motion Control werden in nachstehender Abbildung gezeigt.

Positionieren mit WinAC RTX

Abbildung 3-2



Der wesentlichste Baustein ist der rot gekennzeichnete Achs DB. Er beinhaltet die Achsinformationen jeweils einer Achse. Die blau dargestellten Fahr FB's greifen auf diesen Datenbaustein zu. Auch die Peripheriebausteine und der Regelbaustein (alle gelb dargestellt) nutzen den Achs DB für die jeweiligen Aufgaben. Bei den Peripheriebausteinen handelt es sich um vorkonfektionierte Bausteine zur einfachen Anbindung von Ein- und Ausgangsperipherie.

Achs DB (DB1 Axis)

Der Achs DB ist ein globaler Datenbaustein, auf den alle Easy Motion Control Bausteine zugreifen. Er stellt die zentrale Schnittstelle zwischen den Fahr FB's, dem Lageregler und den Peripheriebausteinen jeweils einer Achse dar. In ihm werden alle Daten dieser Achse gespeichert. Unterschieden werden dabei die Parameterdaten (z. B.: Achsentyp), die aktuellen Daten (z. B.: Lage-Istwert) und die Fehler (bzw. Quittierungen).

Sollen mehrere Achsen verwendet werden, so müssen auch mehrere Achs DB's erstellt werden.

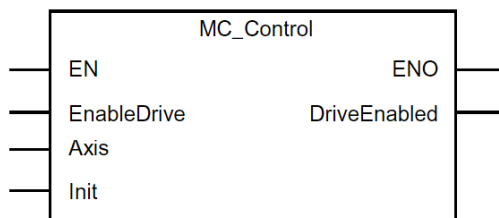
Easy Motion Control Regler (FB11 MC_Control)

Der Lageregler der EMC Software regelt die von den Fahr FB's vorgegeben Werte aus. Er greift dabei auf den jeweiligen Achs DB zu (oben als gelber Baustein dargestellt).

Das folgende Bild stellt den Baustein im FUP/KOP dar.

Positionieren mit WinAC RTX

Abbildung 3-3



Parameter des FB11, MC_Control

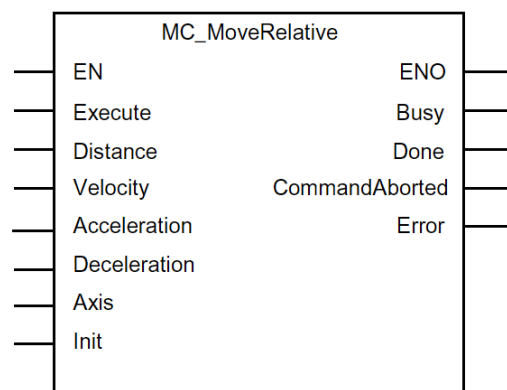
Tabelle 3-1

Eingang/Ausgang	Funktion
Axis	Mit den beiden Eingängen Axis wird die Verbindung zur jeweiligen Achse (durch Achs DB) hergestellt. Jeder EMC Funktionsbaustein besitzt diese beiden Eingänge.
Init	Init stellt ein Initialisierungsbit des jeweiligen Achs-DB's bereit (jeder Baustein benötigt ein exklusives, nur von ihm verwendetes Bit).
EnableDrive	Freigabe der Achse
DriveEnabled	Mit dem Ausgang DriveEnabled wird der Zustand des Reglers zurückgegeben. Wenn kein Fehler anliegt, wird dieses Bit gesetzt.

Easy Motion Control Fahr FB's (FB1...FB5)

Mit den Fahr FB's werden die speziellen Funktionalitäten der Positionierung umgesetzt. Sie greifen wiederum auf den Achs DB zu. Jeder Fahr FB besitzt eine feste Fahrfunktion (z.B.: MC_MoveRelative zur relativen Positionierung, oben als blaue Bausteine dargestellt). Der nachfolgende Baustein zeigt stellvertretend für alle Fahr FB's den Aufbau des Bausteins MC_MoveRelative. Der Aufbau der anderen Bausteine ist ähnlich.

Abbildung 3-4



Positionieren mit WinAC RTX

Parameter der Fahr FB's (FB1...FB5)

Tabelle 3-2

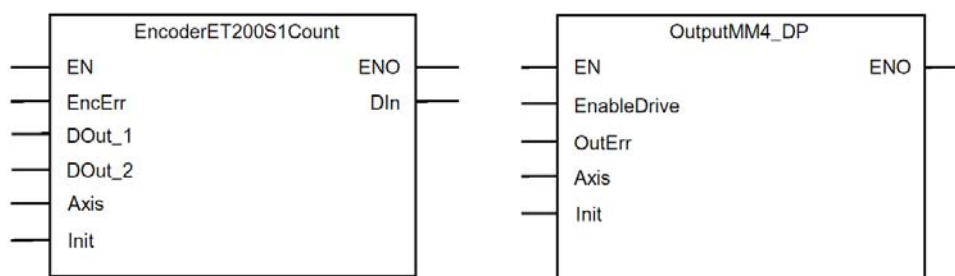
Eingang/Ausgang	Funktion
Velocity, Acceleration, Deceleration	Vorgabe der Geschwindigkeit, Beschleunigung und Bremsverzögerung. In unserem Beispiel wurden diese Werte mit den Maximalwerten der Achse (aus dem Achs-DB) parametrisiert.
Distance	Angabe der zu fahrenden relativen Entfernung zum aktuellen Punkt.
Execute	Starten des Fahrauftrages.
Busy	Der Ausgang Busy wird gesetzt, wenn der Fahrauftrag gegenwärtig ausgeführt wird.
Done	Anzeige, dass Auftrag ausgeführt wurde. (Wird nur kurz gesetzt).
CommandAborted, Error	Fehlermeldungen (siehe Handbuch Easy Motion Control)
Axis	Siehe oben (MC_Control).
Init	Siehe oben (MC_Control).

Eingangs- und Ausgangstreiber (FB 26 und FB 37)

Mit den vorkonfektionierten Ein- und Ausgangstreibern können die Daten auf einfache Art in Easy Motion Control verarbeitet werden. Diese Treiber liegen wieder als Funktionsbausteine vor und können je nach Bedarf in ein STEP 7 Projekt eingebunden werden. Die nachfolgend aufgeführten beiden Treiberbausteine stellen die in unserem Beispiel verwendeten Ein- und Ausgangstreiber dar. Verwendung finden die beiden Bausteine im:

- Eingangstreiber EncoderET200S1Count (für Countermodul in einer ET200S) und
- Ausgangstreiber OutputMM4_DP (PROFIBUS Anschaltbaugruppe für MICROMASTER 440).

Abbildung 3-5



Parameter der Eingangs- und Ausgangstreiber (FB 26 und FB 37)

Tabelle 3-3

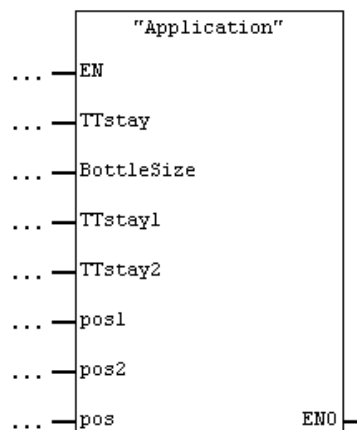
Eingang/Ausgang	Funktion
Axis	Siehe oben (MC_Control).
Init	Siehe oben (MC_Control).
EnableDrive	Freigeben der Ausgangsperipherie. Dieser Eingang wird mit dem Ausgang DriveEnabled des Reglerbausteins MC_Control verschaltet.
DOut_1, DOut_2, DIn	Digitalein- und ausgänge (bleiben im Beispiel frei und werden vom Easy Motion Control intern angesprochen)
EncErr, OutErr	Fehlerbehandlungs-Ein- und Ausgänge (werden nicht benötigt). Für detaillierte Erläuterungen finden Sie im Handbuch Easy Motion Control.

Positionieren mit WinAC RTX

Anwendungsbaustein FB 111

Der Anwendungsbaustein FB111 beinhaltet eine Schrittkette, welche die Achse initialisiert, positioniert und wieder in den Ruhezustand fährt. Zur Realisierung der Positionieralgorithmen greift der FB111 direkt auf die Datenbausteine der jeweiligen Fahr FB's zu.

Abbildung 3-6



Parameter des Anwendungsbausteins FB 111

Tabelle 3-4

Eingang/Ausgang	Funktion
TTstay	Angabe der aktuell gewählten Verweilzeit zum Abfüllen des Getränkes.
BottleSize	Mit der Variable BottleSize ist eine schnelle Auswahl der Flaschenart möglich (0 = kleine Flasche, 1 = große Flasche).
TTstay1, TTstay2, Pos1, Pos2	TTstay1, TTstay2, Pos1 und Pos2 geben jeweils die Verweilzeit und den zu fahrenden Weg für die kleinen und großen Flaschen an. (1 = klein, 2 = groß).

Der Parameter TTStay gibt die aktuell gewählte Verweilzeit zum Abfüllen des Getränkes an. Mit der Variable BottleSize ist eine schnelle Auswahl der Flaschenart möglich (0 = kleine Flasche, 1 = große Flasche).

Durch pos wird der aktuell relativ zu fahrende Weg definiert.

TTstay1, TTstay2, Pos1 und Pos2 geben jeweils die Verweilzeit und den zu fahrenden Weg für die kleinen und großen Flaschen an. (1 = klein, 2 = groß).

Positionieren mit WinAC RTX

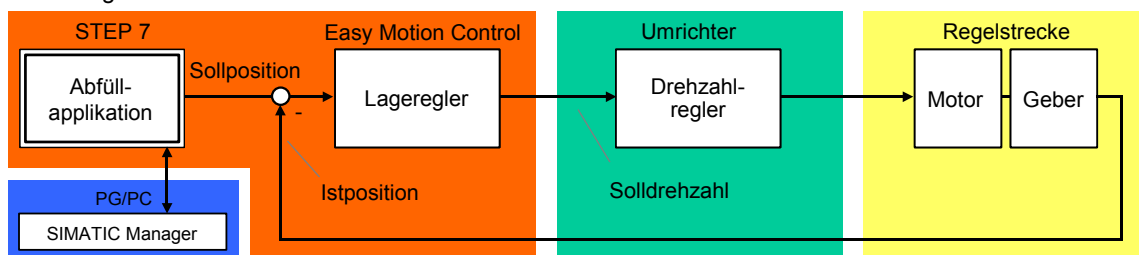
3.5 Logische Kernstruktur im Beispiel

In diesem Beispiel wird auf einfache Art aufgezeigt, wie mit Hilfe der Software SPS WinAC V4.1 eine Lageregelung verwirklicht werden kann. Dazu soll der Motor des Förderbandes für die abzufüllenden Flaschen jeweils eine Flasche unter der Abfülldüse positionieren.

Datenflussmodell

Die folgende Grafik veranschaulicht die in diesem Beispiel wichtigen Datenflüsse.

Abbildung 3-7



Die Bedeutung der Parameter

Die **Sollposition** gibt die vom Anwenderprogramm gewünschte Position an. Sie kann sowohl relativ als auch absolut übermittelt werden.

Als **Solldrehzahl** wird die für den Motor nötige Drehzahl bezeichnet.

Mit der **Istposition** wird die derzeitige Position der Achse ermittelt und bildet die Grundlage zur Neuberechnung der Solldrehzahl.

Zusätzlich sind noch die Parameter **Geschwindigkeit, Beschleunigung und Achsverzögerung** an den Lageregler übertragbar.

Beschreibung der Abläufe im Beispielpogramm

In der nachfolgenden Tabelle wird der zu Grunde liegende Ablauf des Beispiels beschrieben.

Tabelle 3-5

Nr.	Aktion
1	Initialisierung der Achse
2	Freigabe der Achse
3	Übermittlung der Sollposition an den Lageregler
4	Berechnung der Solldrehzahl
5	Einlesen der Istposition der Achse vom Geber
6	Berechnung der Abweichung der Ist- von der Sollposition
7	Neue Solldrehzahl an Umrichter übermitteln
8	Wiederholung ab Schritt 5, solange Sollposition erreicht
9	Weiter bei Schritt 3

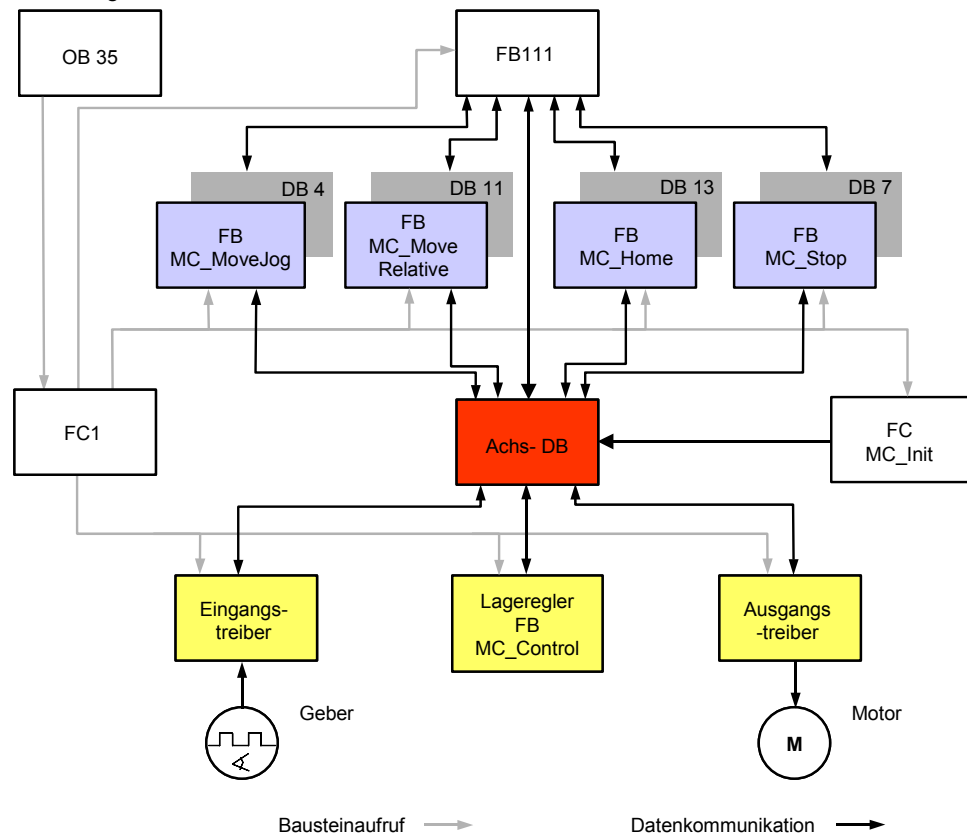
3.6 Programmstrukturen

Im folgenden Kapitel werden der Aufbau und die Struktur des Beispiels auf Funktions- und Datenbausteinebene des Automatisierungssystems betrachtet.

Bausteinstruktur

Nachfolgendes Bild zeigt die Bausteinstruktur des S7-Programms auf den beiden Steuerungen.

Abbildung 3-8



Beschreibung der Bausteinstruktur

In diesem Beispiel wird auf eine komplexe Strukturierung verzichtet, um schnell die grundsätzlichen Funktionalitäten verdeutlichen zu können. Die grobe Bausteinstruktur kann folgendermaßen charakterisiert werden:

- Im Weckalarm OB 35 werden zyklisch die Easy Motion Control Bausteine und der Funktionsbaustein FB111 aufgerufen.
- Der FB111 enthält die Schrittkette zur Realisierung der Abfüllanwendung.
- Im FB111 wird in Abhängigkeit vom aktuellen Schritt auf die Datenbausteine der Fahr FB's sowie auf den Achs DB zugegriffen. Über die Datenbausteine der Fahr FB's werden deren Funktionen aufgerufen.

3.7 Programmablauf

Nachfolgend wird das S7-Programm beschrieben, das die Abfüllanwendung in der WinAC bereitstellt. Grundsätzlich wird das gesamte Easy Motion Control Programm in einem Weck Alarm OB ausgeführt (hier OB35).

Programmablaufplan

Nachfolgend wird das S7-Programm beschrieben, das die Abfüllanwendung in der WinAC bereitstellt. Grundsätzlich wird das gesamte Easy Motion Control Programm in einem Weck Alarm OB ausgeführt (hier OB35).

Tabelle 3-6

Flussdiagramm	Beschreibung
<pre> graph TD Start([OB 35 Zyklus START]) --> FC[Bedingter Aufruf FC MC_Init] FC --> FB1[Aufruf FB EncoderET200S1Count] FB1 --> FB2[FB 111 Applikation] FB2 --> FB3[Aufruf FB MC_MoveJog] FB3 --> FB4[Aufruf FB MC_Relative] FB4 --> FB5[Aufruf FB MC_Home] FB5 --> FB6[Aufruf FB MC_StopMotion] FB6 --> FB7[Aufruf FB OutputMM4_DP] FB7 --> End([OB 35 Zyklus ENDE]) </pre>	Abhängig vom Zustand der Anlage (bei Neuanlauf, Reset oder Fehler) wird eine Achseninitialisierung vorgenommen. Einlesen der Geberwerte über die ET200S. Aufruf des Applikationsbausteins (Ausführung der Schrittkette) Bereitstellung der Jog Funktionen. Bereitstellung der Relativbewegungen. Bereitstellung der Nullpunktsynchronisationsfunktionen. Aufruf der eventuellen Stop Funktion. Ansteuern der Motorperipherie.

Positionieren mit WinAC RTX

Schrittfolge des Anwenders Bausteins FB111 (Application)

Zur Ausführung der Abfüllapplikation im Baustein FB 111 wurde eine Schrittfolge in STEP 7 programmiert, welche sich wie folgt zusammensetzt:

Tabelle 3-7

Nr.	Name des Schrittes	Aktion
1	SInit	Initialisierungsschritt
2	SAxisFree	Achsfreigabe
3	SErrorAck	Fehlerquittierung nach Initialisierung
4	SHpos	Positionierung von Hand
5	SHome	Bezugspunkt setzen
6	SInitFilling	Initialisierung des Füllvorganges (Zeit und Positionierung werden in Abhängigkeit von der Flaschengröße in die Variablen TTstay und pos übertragen)
7	SFill	Abfüllschritt (Simulation des Abfüllvorganges mit einem Timer)
8	SPlacing	Positionierung mit dem Baustein MC_MoveRelative. Das Band wird um eine vorgegebene Entfernung (in Abhängigkeit von der Flaschengröße) nach vorn bewegt.
9	SAxisRun	Achse befindet sich in Bewegung
10	SAxisReady	Achse ist am Ziel angekommen

Positionieren mit WinAC RTX

Codeauszug des Applikationsbausteins mit der Positionieranforderung

Der folgende Codeauszug aus dem FB111 zeigt den Schritt acht der Schrittkette, welcher die Positionierung der Achse in Form eines Datenbausteinzugriffes realisiert.

Netzwerk 8: Schritt 8 / stepp 8 (SPlacing)

```

U T1                                     // Wenn Timer 1 abgelaufen ist
                                     // If timer 1 has gone off
S "SPlacing"                             // aktiviere Schritt 8 (Positionieren)
                                     // activate step 8.

U(
  O "DB_MC_MoveRelative".Done           // Rücksetzung erfolgt über Schritt 10
  O "Halt"                               // The step 8 is resetting by the step 10
  O "Reset"                              // oder Halt, / the Reset or the Halt flag.
)                                         // bzw. Reset
R "SPlacing"
U "SPlacing"                             // Wenn Schritt 8 (Positionieren) aktiv ist
= L 0.0                                  // If step 8 is active
U L 0.0                                  // dann lade die relative Fahrweite in den Datenbau-
SPBNB _005                               // stein des FahrFB's MC_MoveRealative
L #pos                                   // then transfer the relative move wide into the
T "DB_MC_MoveRelative".Distance         // move FB MC_MoveRealative
_005: NOP 0                             // und setze den Eingang Execute am DB zum Starten
U L 0.0                                  // der Positionierung
BLD 102                                  // and set the execute flag at the FB
= "DB_MC_MoveRelative".Execute          // MC_MoveRealative

```

Beschreibung des Positionieraufufes

Nach dem Schritt 7 (Abfüllen /SFilling) und Beenden des Timers wird der Schritt 8 gestartet. In ihm wird die relative Distance zum nächsten Punkt in den Datenbaustein des FB_MoveRelative geschrieben. Auf die gleiche Weise wird der Fahrvorgang dann mit Hilfe des Execute Bit's im Datenbaustein gestartet.

Positionieren mit WinAC RTX

Codeauszug des OB35 mit dem Aufruf des Fahr FB's MC_MoveJog

Der folgende Codeauszug stellt den Aufruf des Fahr FB's MC_MoveJog im OB35 sowie dessen Parameterübergabe dar.

Netzwerk 4: Handbetrieb / manual funktion

```
CALL "MC_MoveJog" , "DB_MC_MoveJog" // Aufruf des TIPP-Funktionsbausteins MC_MoveJog
                                     // Call of the Jog FB.

JogPos      :=
JogNeg      :=
Velocity    := "Axis".Ax.MaxVelocity // Setzen der Geschwindigkeit auf die maximal zu-
                                     // lässige Achsgeschwindigkeit
                                     // Set the velocity to the most permitted axis
                                     // velocity.
Acceleration := "Axis".Ax.MaxAcceleration // Setzen der Beschleunigung auf die
                                     // maximal zulässige Achsbeschleunigung
                                     // Set the acceleration to the most permitted axis
                                     // acceleration.
Deceleration := "Axis".Ax.MaxDeceleration // Setzen der Verzögerung auf die maxi-
                                     // mal zulässige Achsverzögerung
                                     // Set the deceleration to the most permitted axis
                                     // deceleration.

Busy        :=
Done        :=
CommandAborted:=
Error       :=
Axis        := "Axis".Ax           // Zuweisung der Achse
                                     // Assignment of the axis
Init        := "Axis".Ax.Init.I2   // Zuweisung des Initialisierungsbits
                                     // Assignment of the initialization flag.

NOP         0
```

Beschreibung des Positionieraufufes

Das Codebeispiel zeigt exemplarisch den Aufruf des Positionierbausteins für den TIPPbetrieb (manuelle Achsbewegung). Wie bereits oben erwähnt, wurden für die Bewegungsparameter Velocity, Acceleration und Deceleration die maximal zulässigen Werte der Achse aus dem Achs DB übergeben.

Mit Axis wird dem Funktionsbaustein die zu verwendende Achse mit Hilfe des dazugehörigen Achs DB's DB1 („Axis“) übergeben.

Der Variable Init wird ein Initialisierungsbit des Achs DB's zugewiesen. Dabei ist wichtig, dass dieses Bit nur einmal vergeben wird. Jeder Easy Motion Control Funktionsbaustein besitzt sein eigenes Initialisierungsbit.

Es dürfen keine Initialisierungsbits doppelt vergeben werden!

Alle anderen Fahr FB's sind auf die gleiche Weise parametrisiert.

4 Installation der Hard- und Software

Dieses Kapitel befasst sich mit der Installation der Komponenten für die Applikation. Es gliedert sich in folgende Abschnitte.

Behandelte Themen

Kap.	Titel	Seite
4.1	Vorbereitende Installation	29
4.2	Hardwareaufbau	31
4.3	Installation der WinAC RTX Software	40
4.4	Benutzung von WinAC RTX	40
4.5	Konfiguration der Automatisierungsstationen	40

Installationsreihenfolge

Folgende Tabelle zeigt die Reihenfolge für das Inbetriebnehmen der Applikation sowie wichtige Installations- und Aufbaureihenfolgen.

Tabelle 4-1

Nr.	Aktion
1	STEP 7 V5.2 SP1 (oder höher)
2	SIMATIC NET PC Software V6.1 SP1 (oder höher)
3	Installation der Parametriersoftware STARTER oder Drive ES
4	Installation des CP5613 A2
5	Installation der Hardware für die Station 2 (ET200S) und Station 3 (MIC-ROMASTER 440)
6	WinAC RTX V4.1 (oder höher) Diese Installation teilt sich in folgende Schritte auf: • Installieren und Überprüfen der VenturCom RTX-Erweiterungen • Installieren und Autorisieren der Software WinAC RTX

4.1 Vorbereitende Installation

STEP 7

Die Installation von STEP 7 erfolgt auf dem PG/PC, welches für die Projektierung und Programmierung der Automatisierungsstationen vorgesehen ist. Alternativ dazu kann STEP 7 auch auf dem PC (Station 1) installiert werden, auf dem WinAC laufen soll.

Auf die Beschreibung der Installation von STEP 7 wird an dieser Stelle verzichtet. Die Installation findet in gewohnter Windows-Umgebung statt und ist selbsterklärend.

Positionieren mit WinAC RTX

SIMATIC NET PC Software

Die SIMATIC NET PC Software wird auf dem PC (Station 1) installiert, auf welchem anschließend auch die WinAC installiert werden soll. Das Softwarepaket enthält alle benötigten Hilfsmittel zur Einrichtung und zum Betrieb einer PC-Station.

Ab der STEP 7 Version 5.2 wird zur Inbetriebnahme von PC-Stationen das Verfahren „Advanced PC Configuration“ benutzt. Dieses ermöglicht die Projektierung von PC-Stationen direkt in STEP 7. Vor der Benutzung von „Advanced PC Configuration“ empfiehlt es sich unbedingt, das Handbuch /2/ „SIMATIC NET, PC-Stationen in Betrieb nehmen – Anleitung und Schnelleinstieg“ zu lesen.

Starter Installation

Zur Parametrierung des Umrichters ist die Software „Starter“ oder alternativ dazu das Tool Drive ES notwendig.

Installieren Sie bitte eines der beiden Pakete auf ihrem PG/PC.

Die Installation erfolgt in der in Windows üblichen Setup Abfolge.

Installation der Easy Motion Control Software

Zur Parametrierung der Achse und zum Programmieren benötigen Sie die Easy Motion Control Software.

Die Installation der Software erfolgt in der in Windows üblichen Setup Abfolge.

Hinweis

Da die Easy Motion Control Bausteine lizenzpflichtig sind, werden sie nicht im Projekt mitgeliefert. Das Projekt ist nur ausführbar, wenn Easy Motion Control installiert wurde.

Positionieren mit WinAC RTX

4.2 Hardwareaufbau

Der grundsätzliche Hardwareaufbau ist in Kapitel 2.1 zu sehen.

4.2.1 Industrie PC / WinAC (Station 1)

Informationen zum Hardwareaufbau und zur Inbetriebnahme des Industrie PC's entnehmen Sie bitte der Applikation „Grundlagen für Lösung von Automatisierungsaufgaben auf Basis von WinAC RTX“ (Beitrags-ID: [21004765](#)).

4.2.2 ET200S Station (Station 2)

Hardwareaufbau der ET200S Station (Station 2)

Tabelle 4-2

Nr.	Aktion
1	Die Stromversorgung wird mit Hilfe des Hutschienenadapters auf der Hutschiene montiert.
2	Das Interfacemodul wird neben die Stromversorgung auf die Hutschiene gesetzt.
3	Anschließend werden die beiden Terminalmodule und auf ihnen jeweils das Power- und Countermodul installiert. Das Powermodul wird links vom Countermodul aufgesteckt. Eine detaillierte Installationsanleitung erhalten Sie im Handbuch ET 200S (Beitrags-ID: 1144348) und ET 200S Technologische Funktionen (Beitrags-ID 9264111).
4	Versorgen Sie das Interface Modul mit der Betriebsspannung.
5	Verschalten Sie das Powermodul wie folgt mit der Stromversorgung. - Pin 2 mit +24V DC (L+) - Pin 3 mit 0 V DC (M)
6	Stellen Sie am Interfacemodul mit Hilfe der DIL Schalter die Adresse 7 ein. (Bit 1,2 und 3 Setzen)
7	Verbinden Sie die ET200S (Station 2) mit Hilfe eines PROFIBUS-Kabels mit dem Industrie PC (Station 1). Achten Sie auf die Abschlusswiderstände.
8	Schließen Sie den Geber mit ausreichender Kabellänge wie in der folgenden Tabelle dargestellt an die Counter-Baugruppe an.

Verdrahtung der ET200S Station (Station 2)

Tabelle 4-3

Signal	24V Geber (Steckerbezeichnung)	Leitungsfarbe	1Count
U _{a1}	E	gelb	Kanal A (Pin 5 TM)
U _{a2}	H	lila	Kanal B (Pin 1 TM)
0V (M)	K	grau	0V (Pin 7 TM)

Positionieren mit WinAC RTX

U _p (24V DC)	M	schwarz/weiß	24V (Pin 6 TM)
-------------------------	---	--------------	----------------



Achtung

Die Verdrahtungstabelle muss in jedem Fall mit der vorhandenen Hardware abgestimmt werden. Prüfen Sie bitte vor der Inbetriebnahme sorgfältig an Hand des Handbuches, ob die oben angegebene Verdrahtung für ihre Hardware zulässig ist, da es sonst zu Fehlfunktionen oder Zerstörung der Hardware kommen kann!

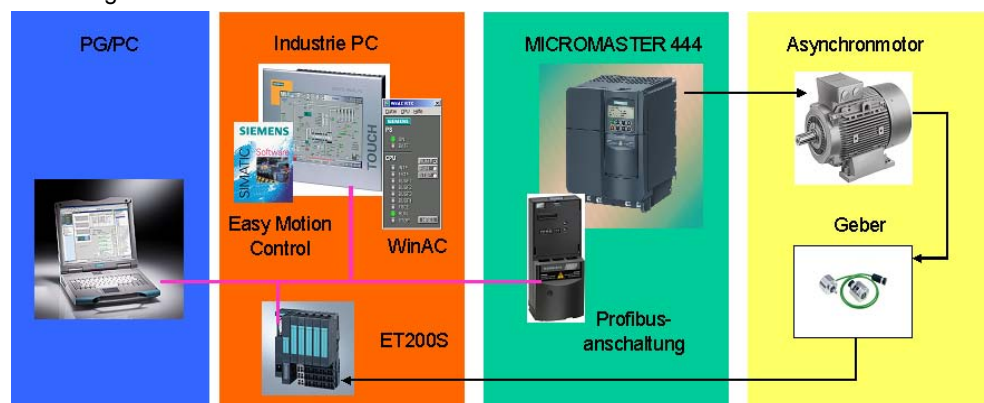
Inbetriebnahme der ET200S Station (Station 2)

Nach dem Zuschalten der Spannung sollte die ON LED des Interfacemodules leuchten. Beim Powermodul sollte die ON LED leuchten.

4.2.3 MICROMASTER 440 (Station 3) und Motor

Verdrahtungsschema (gesamt)

Abbildung 4-1



Verdrahtungstabelle MICROMASTER und Motor

Tabelle 4-4

Signal	MM 440	Motor
230V AC-P	L	
230V AC-N	N	
PE	PE	PE
Umrichter Ausgang	U	U1
Umrichter Ausgang	V	V1
Umrichter Ausgang	W	W1

Positionieren mit WinAC RTX

Aufbau MM 440 und Motor

Tabelle 4-5

Nr.	Aktion
1	Schließen Sie den MICROMASTER 440 , wie in der Verdrahtungstabelle dargestellt, an die Spannungsversorgung an.
2	Verbinden Sie nun den MICROMASTER , wie oben angegeben, mit dem Motor.
3	Schalten Sie den Motor im Klemmkasten in Dreieckschaltung . (siehe Betriebsanleitung MICROMASTER 440)
4	Installieren Sie die PROFIBUS-Anschaltung , indem Sie die Optionsbaugruppe auf den MICROMASTER aufsnappen.
5	Stellen Sie die Adresse der Baugruppe mit Hilfe der DIL Schalter auf den Wert 15 (Bit 1,2,3 und 4 auf ON Schalten).
6	Verbinden Sie den MICROMASTER 440 (Station 3) mit Hilfe eines PROFIBUS-Kabels mit dem Industrie PC (Station 1). Achten Sie auf die Abschlusswiderstände .
7	Falls am PG keine PROFIBUS-Schnittstelle vorhanden und die SIMATIC Steuerung nicht routingfähig ist, kann zusätzlich noch der PC_Umrichter-Verbindungssatz auf die PROFIBUS-Optionsbaugruppe aufgeschnappt werden.
8	Installieren Sie den Geber , wie im mitgelieferten Handbuch dargestellt, am Motor.

Parametrierung und Inbetriebnahme des MICROMASTERS mit STARTER

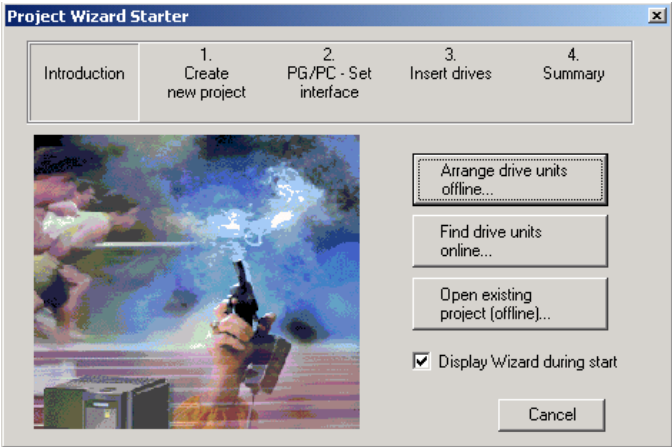
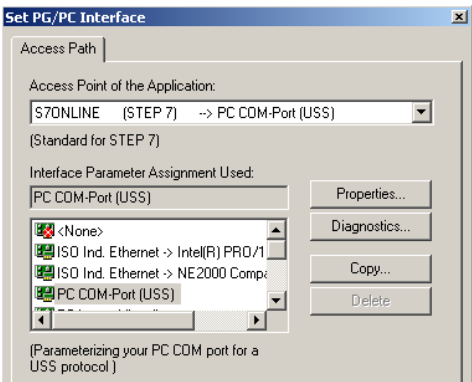
Da die Parametrierung mit Hilfe von Drive ES der Parametrierung mit STARTER zu großen Teilen gleicht, wird hier nur auf die Vorgehensweise mit STARTER eingegangen.

Verfügt Ihr PG/PC über eine PROFIBUS-Schnittstelle, kann die Parametrierung des Umrichters über den PROFIBUS vorgenommen werden. Alternativ dazu benötigen Sie eine serielle Schnittstelle und das PC_Umrichter-Verbindungsset, über die Sie sich mit dem MICROMASTER verbinden können.

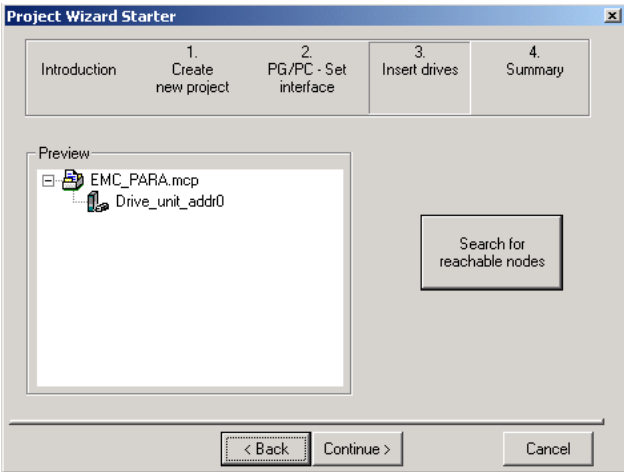
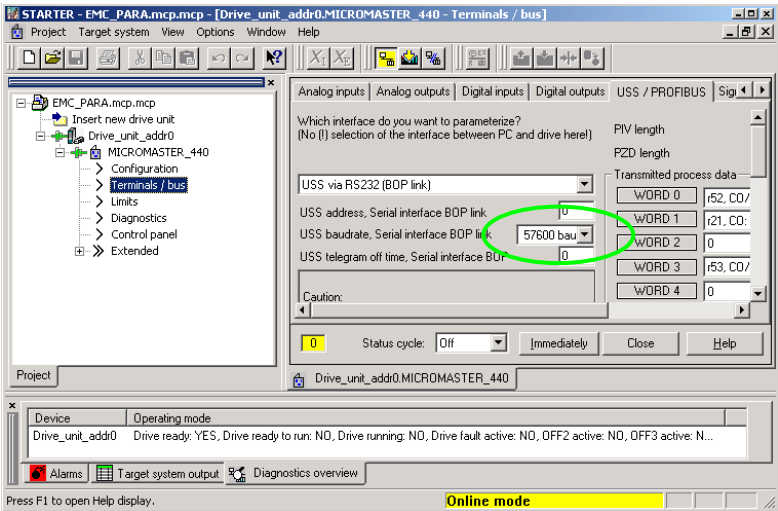
Tabelle 4-6

Nr.	Aktion
1	Verbinden Sie Ihr PG über den PROFIBUS mit der MICROMASTER PROFIBUS Optionsbaugruppe. Alternativ über die serielle Schnittstelle (mit dem PC_Umrichter-Verbindungsset).
2	Starten Sie das Programm STARTER (Doppelklick auf Desktopsymbol  oder unter „ START → SIMATIC → STARTER “)

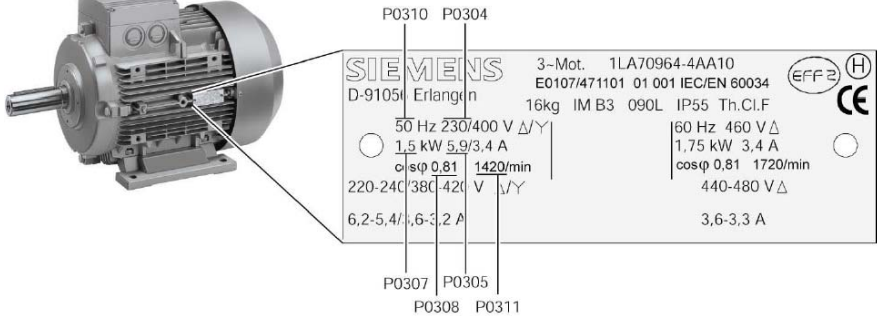
Positionieren mit WinAC RTX

3	 Sollte der Assistent nicht automatisch angezeigt werden, können Sie ihn im Menü „Projekt → Neu mit Assistent“ aufrufen.
4	Wählen Sie „ Antriebsgeräte online suchen... “
5	Geben Sie den gewünschten Projektnamen ein. Auf Wunsch können Sie auch das Ablageverzeichnis ändern und die anderen Felder ausfüllen. Beenden Sie den Schritt mit „Weiter“.
6	Klicken Sie auf Ändern und testen... , um die Schnittstelle auszuwählen.
7	 Falls Sie den PC – Umrichter Verbindungssatz verwenden, wählen Sie „PC COM-Port (USS)“ und stellen Sie mit „Eigenschaften“ die auf Sie zu- treffenden Kommunikationsparameter ein. Unter Baudratentest können Sie ermitteln, welche Baudrate am MICRO- MASTER eingestellt ist. Stellen Sie die gleiche Baudrate für die Schnittstel- le ein wie am Umrichter ermittelt wurde. Für die Verwendung der PROFIBUS-Schnittstelle stellen Sie bitte die PROFIBUS Schnittstelle im Dialogfeld ein. Bestätigen Sie mit „Ok“ und klicken Sie anschließend auf „Weiter“.

Positionieren mit WinAC RTX

8	Klicken Sie auf „Erreichbare Teilnehmer suchen“.  Wählen Sie „Antriebsgerät_Adr0“ aus, klicken Sie auf „Weiter“ und anschließend auf „Fertigstellen“.
9	Gehen Sie online durch Klicken auf diesen Button. 
10	Stellen Sie die Werkseinstellungen wieder her durch rechten Mausklick auf „Antriebsgerät_Adr0 → Zielgerät → Werkseinstellungen wiederherstellen“. Bestätigen Sie mit Ok.
11	Verwenden Sie den PC-Umrichter-Verbindungssatz, empfiehlt es sich, die Baudrate herauf zu setzen. • Klicken Sie doppelt auf „Klemme/Bus“ im Baumdiagramm • Wählen sie im Reiter „USS/PROFIBUS“ als zu parametrierende Schnittstelle „USS über RS232 (BOP-link)“ aus • Stellen Sie nun „57600 Baud (9)“ ein • Führen Sie die aufgeblendeten Anweisungen durch (trennen, PC/PG Schnittstelle neu einstellen, wieder verbinden)  Hinweis Möchten Sie anschließend wieder ein AOP verwenden, muss die Baudrate zunächst wieder auf 9600 eingestellt werden.

Positionieren mit WinAC RTX

12	Klicken Sie doppelt auf „MICROMASTER_440 → Konfiguration“. Wählen Sie nun „Antrieb neu konfigurieren...“.
13	Der gestartete Assistent fragt nun die folgenden Motor- und Umrichterdaten ab. Sie finden die Informationen auf dem Typenschild des Motors. Daten, welche hier nicht behandelt werden, lassen Sie bitte unverändert. • Wählen Sie „konstantes Moment“ • Stellen Sie als Motortyp „Asynchronmotor“ ein. • Geben Sie die Motordaten laut Typenschild ein. Auf dem Typenschild nicht vorhandene Informationen bleiben unverändert. • Bei „Konfiguration Impulsgeber“ wählen sie „Gesperrt (0)“ • Als „Betriebsmodus“ wählen Sie die „Vektorregelung ohne Sensor (20)“ • Die „Steuersignalherkunft“ wird auf „CB an COM – Link (6)“ eingestellt • „Drehzahlollwerte“ werden ebenfalls von „CB an COM – Link (6)“ empfangen.  Das dargestellte Motortypenschild zeigt folgende Parameter: - P0304 = Nennspannung (in V) - P0305 = Nennstrom (in A) - P0307 = Nennleistung (in kW) - P0311 = Nenndrehzahl (in U/min) - P0310 = Nennfrequenz (in Hz) - P0308 = Cos(phi) / Nennleistungsfaktor
14	Stellen Sie bei der „Hochlaufzeit“ und den BEIDEN Rücklaufzeiten 0 Sekunden ein!
15	Klicken Sie auf „Fertigstellen“ und bestätigen Sie die Sicherheitsabfrage.

Positionieren mit WinAC RTX

16

Führen Sie nach der Eingabe der Motordaten eine Motoridentifikation durch. Der MICROMASTER misst dabei die für die Vektorregelung wichtigen Daten, welche nicht auf dem Typenschild vorhanden sind, wie z.B.: Kabellängen usw.

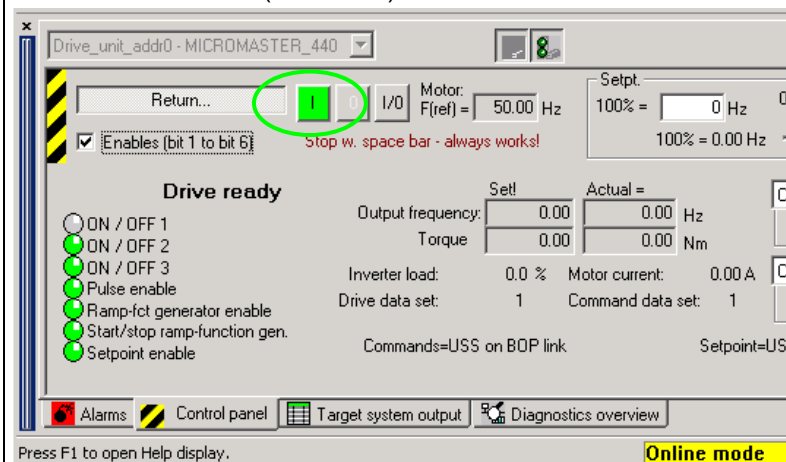
Hinweis

Durch die Motoridentifikation wird auch das Temperaturmodell des Motors im Umrichter optimiert, welches vor Überlast schützt. Führen Sie diesen Schritt unbedingt durch, da der Motor gerade bei langsamen Positionierungen sehr schnell hohe Temperaturen erreicht (geringe Eigenkühlung)!

Achtung

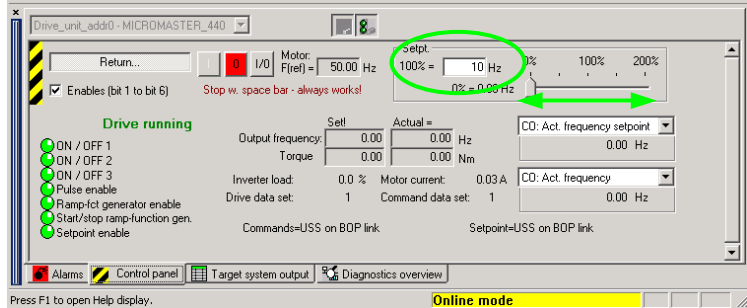
Durch die Motoridentifikation wird der Motor eingeschaltet und kann sich gegebenenfalls drehen!

- **Klicken** Sie nun auf „**Motoridentifikation**“.
- Beachten Sie die Hinweise.
- Rufen Sie die Steuertafel durch einen **Doppelklick** auf „**Steuertafel**“ in der Baustruktur auf.
- **Klicken** Sie auf „**Steuerhoheit holen**“, beachten Sie die Hinweise.
- Setzen Sie einen **Haken** bei „**Freigaben (Bit 1 bis Bit 6)**“
- Und nun auf **Ein** (siehe Bild).

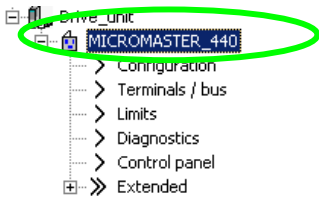


- Warten Sie bis in der Steuertafel „**Einschaltbereit**“ erscheint!
- Nach Fertigstellung der Identifikation **drücken** Sie bitte die **Leertaste** (oder Klicken wieder auf den **Aus (0) Button**)
- Geben Sie die Steuerhoheit nun wieder zurück durch Klicken auf den Button **...zurückgeben**.

Positionieren mit WinAC RTX

17	Führen Sie nun noch eine Sättigungsermittlung durch. (Sie benötigen wieder die Steuerhoheit) Achtung Durch die Sättigungsermittlung wird der Motor eingeschaltet und kann sich gegebenenfalls drehen! • Klicken Sie auf „Ermittlung der Sättigung“. • Beachten Sie die Hinweise! • Holen Sie sich die Steuerhoheit für den Antrieb wie oben beschrieben. • Schalten Sie den Motor in der Steuertafel, wie oben beschrieben, wieder zu. • Warten Sie, bis in der Steuertafel „Einschaltbereit“ erscheint! • Geben Sie die Steuerhoheit nun wieder zurück durch Klicken auf den Button „...zurückgeben“. Hinweis Alternativ zur Steuertafel lassen sich die Motoridentifikation und die Sättigungsermittlung auch mit zugeschalteter CPU und geladenem Beispielprogramm starten. Setzen Sie dazu die Schritt看ette zurück und anschließend das Initialisierungsbit (M0.0) auf eins. Detaillierte Informationen zur Bedienung des Beispielprogrammes erhalten Sie in Kapitel 7.
18	Nach den automatischen Messungen des Umrichters können Sie nun einen Testlauf des Motors vornehmen. Gehen Sie dazu wie folgt vor: Achtung Durch das Testen des Motors sind verschiedene Sicherheitsbestimmungen Ihrerseits einzuhalten. Beachten Sie unbedingt die eingeblendeten Sicherheitshinweise! • Holen Sie sich die Steuerhoheit für den Antrieb, wie oben beschrieben. • Schalten Sie den Motor in der Steuertafel, wie oben beschrieben, wieder zu. 

Positionieren mit WinAC RTX

18	- Sie können nun bei „Sollwert“ einen Sollwert für die Frequenz des Motors festlegen. Stellen Sie zunächst einen geringen Wert ein (z.B.: 10 Hz). - Mit dem Schieberegler können Sie nun die auszuführenden konkreten Drehzahlen steuern. - Nach den von Ihnen durchgeführten Tests schalten Sie den Motor durch Klick auf den „Button 0“ (rot) wieder aus. Geben Sie die Steuerhoheit nun wieder zurück durch Klicken auf den Button ...zurückgeben.
19	Speichern Sie das Projekt im Rom des Antriebes durch Auswahl des Antriebes  und anschließendem Klicken auf das Symbol  „Ram nach Rom kopieren“. Bestätigen Sie die Sicherheitsabfrage.
20	Laden Sie die Konfiguration nun aus dem Umrichter in das PG. Klicken Sie dazu diesen Button  „in PG laden“ .
21	Speichern Sie nun das Projekt mit dem Schalter  „speichern2“.

4.3 Installation der WinAC RTX Software

Informationen zur Installation der WinAC RTX entnehmen Sie bitte der Applikation „Grundlagen für Lösung von Automatisierungsaufgaben auf Basis von WinAC RTX“ (Beitrags-ID: [21004765](#)).

4.4 Benutzung von WinAC RTX

Informationen zur Benutzung der WinAC RTX entnehmen Sie bitte dem Kapitel 5 der Applikation „Grundlagen für Lösung von Automatisierungsaufgaben auf Basis von WinAC RTX“ (Beitrags-ID: [21004765](#)).

4.5 Konfiguration der Automatisierungsstationen

Informationen zur Konfiguration der WinAC RTX und des Komponentenkonfigurators entnehmen Sie bitte dem Kapitel 6 der Applikation „Grundlagen für Lösung von Automatisierungsaufgaben auf Basis von WinAC RTX“ (Beitrags-ID: [21004765](#)).

Installation des STEP 7-Projektes

Zum Öffnen und Anpassen des STEP 7 Projekts an Ihre Konfiguration gehen Sie wie folgt vor:

Tabelle 4-7

Nr.	Aktion	Hinweis / Erklärung
1	Öffnen Sie den SIMATIC Manager .	
2	Dearchivieren Sie das Archiv „21004767_WinAC_TK_CODE_v10.zip“ über das Menü „Datei > Dearchivieren...“	Nach der Dearchivierung können Sie das Projekt gleich öffnen.
2	Da die Easy Motion Control Bausteine lizenzpflichtig sind, werden sie nicht im Projekt integriert. Öffnen Sie deshalb bitte die Bibliothek „ EMC2 Easy Motion Control “. Öffnen Sie nun das Menü „ Datei “ und wählen Sie „ Öffnen “. Wählen Sie den Reiter „ Bibliotheken “ und dort die Bibliothek „ EMC2 Easy Motion Control “ aus.	

Positionieren mit WinAC RTX

Nr.	Aktion	Hinweis / Erklärung
3	Kopieren Sie nun per Drag und Drop und gedrückter STRG Taste die folgenden Bausteine in den Bausteinbehälter der WinAC RTX des Applikations-Projektes. • FC0 MC_Init • FB2 MC_MoveRelative • FB3 MC_MoveJog • FB4 MC_Home • FB5 MC_StopMotion • FB11 MC_Control • FB26 EncoderET200S1Count • FB37 OutputMM4_DP	Alternativ dazu kann auch der Menübefehl Bearbeiten > Kopieren (Einfügen) verwendet werden.

Parameter der Achse

Zur Parametrierung, zum Test und zur Diagnose (Fehler und Achszustand) der Achsparameter wird die mit Easy Motion Control ausgelieferte Parametriersoftware verwendet. Die Software greift dabei auf den Achs DB zu (DB1 Axis).

Da im STEP 7 Programm für die Geschwindigkeit, die Beschleunigung und die Achsverzögerung jeweils die maximal zulässigen Werte programmiert wurden, soll in folgendem Beispiel kurz die Parametrierung dieser Werte erläutert werden.

Die anderen Werte lassen sich in gleicher Weise ändern.

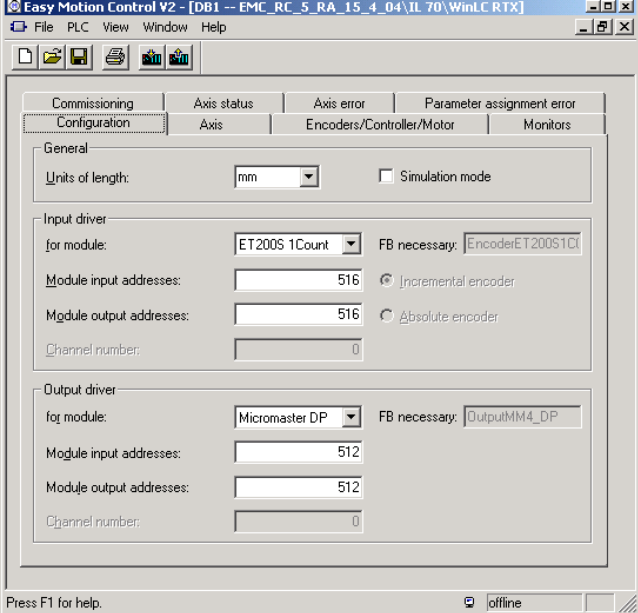
Hinweis Das Umparametrieren der Achse sollte zunächst mit den oben genannten Komponenten nicht nötig sein. Sie sollten die Parameter und Einstellungen nur ändern, falls trotz der im Folgenden beschriebenen Anleitungen keine Positionierung möglich ist.

Genaue Informationen zu den Parametern entnehmen Sie bitte den Handbüchern für Easy Motion Control.

Gehen Sie zum Ändern der Achsparameter folgendermaßen vor:

Positionieren mit WinAC RTX

Tabelle 4-8

Nr.	Aktion
1	Klicken Sie zunächst im Bausteincontainer doppelt auf den Baustein „DB1“. Es öffnet sich die Easy Motion Control Software. 
2	Klicken Sie nun auf den Reiter „ Achse “.
3	Sie können nun die folgenden Parameter einstellen: • Maximale Achsgeschwindigkeit • Maximale Achsbeschleunigung • Maximale Achsverzögerung
4	Klicken Sie nun auf den Button Speichern  und anschließend auf den Button in Steuerung laden .

Hinweis

Je nach verwendetem Motor kann es zu Fehlern beim Fahren der Achse kommen. Sie sollten dann mit den oben genannten Parametern und den Parametern Schleppabstand, Zielbereich und Stillstandsbereich eine Anpassung vornehmen.

Verwenden Sie einen anderen Inkrementalgeber, so müssen Sie unter dem Reiter „Geber/Regler/Motor“ die Anzahl der „Schritte pro Geberumdrehung“ entsprechend anpassen.

Positionieren mit WinAC RTX

Laden des S7-Programms in die WinAC RTX (Station 1)

Das Laden in die WinAC auf Station 1 erfolgt vom PG/PC aus über PROFIBUS. Befolgen Sie dazu die folgenden Schritte.

Tabelle 4-9

Nr.	Aktion
1	Wenn noch nicht geschehen, schalten Sie den PC der Station 1 ein und starten Sie WinAC RTX. Hinweis Achten Sie darauf, dass der Komponenten-Konfigurator korrekt konfiguriert ist.
2	Führen Sie ein Urlöschen in der WinAC RTX Steuerung über den Button  aus.
3	Verbinden Sie Ihr PG mit dem PROFIBUS-CP (CP5613 A2) der Station 1 über ein MPI- bzw. PROFIBUS-Kabel.
4	Zum Einfachen Umschalten der PG/PC-Schnittstelle auf PROFIBUS gehen Sie folgendermaßen vor: • Doppelklicken Sie auf das „PG/PC(1)“ Objekt im STEP 7 Projekt • Ordnen Sie im Bereich „Nicht Zugeordnet“, falls noch nicht geschehen, die PROFIBUS-Schnittstelle über den Button „Zuordnen“ zu. • Selektieren Sie in der Lasche „Zuordnung“ im unteren Bereich „Zugeordnet“ die PROFIBUS-Schnittstelle • Setzen Sie den Haken bei „S7ONLINE-Zugriff:“ • Bestätigen Sie die Eingaben mit „OK“ Damit ist die PG/PC-Schnittstelle auf PROFIBUS umgestellt. Hinweis Bei der Erstkonfiguration des CP5613 A2 als IF-Modul in der Station 1, kann es vorkommen, dass die Baudrate auf 187,5 Kbps eingestellt wird. Passen Sie für das Laden der Parametrierung die Parameter des PROFIBUS-Netzes entsprechend an. Die aktuellen Busparameter eines CP5613 A2 können Sie sich im „WinLC Eigenschaften“-Dialog (aus dem Komponenten Konfigurator heraus) über den Button „Diagnose“ anzeigen lassen (Dazu muss der WinLC RTX Controller gestartet sein.).
5	Öffnen Sie die Hardware Konfiguration der „PCWinAC“ Station und laden Sie diese in die Station 1.
6	Laden Sie nun die Bausteine des S7-Programms der „PCWinAC“ Station in das Zielsystem.
7	Schalten Sie die WinAC RTX Steuerung über den Button  in den Betriebszustand „RUN-P“.

5 Bedienen und Beobachten

Einleitung

Mit der Easy Motion Control Projektierungssoftware steht ein Tool zur einfachen Projektierung, Parametrierung, Beobachtung und Inbetriebnahme der Easy Motion Control Funktionalitäten bereit.

Zur Bedienung und Beobachtung der Beispielapplikation ist im Projekt eine Variablen-tabelle integriert. Mit ihr lassen sich die verschiedenen Funktionen aufrufen und beobachten.

Alternativ zur Variablen-tabelle kann auch das mitgelieferte ProTool / Pro Projekt zur Bedienung und Beobachtung verwendet werden. Dieses finden Sie im gleichen Pfad wie das STEP7 Projekt unter dem Verzeichnis „Pro-Tool Visualisierung“. Da sich die Bedienung der beiden Varianten im Wesentlichen gleicht, wird hier nur auf die Variablen-tabelle eingegangen.

Wird das ProTool Projekt verwendet, muss dazu der Bereich Steuerung an die gegebenen Kommunikationsumstände angepasst werden.

Voraussetzung für das Bedienen und Beobachten ist die Installation der Hard- und Software, welche im Kapitel 4 beschrieben wurde.

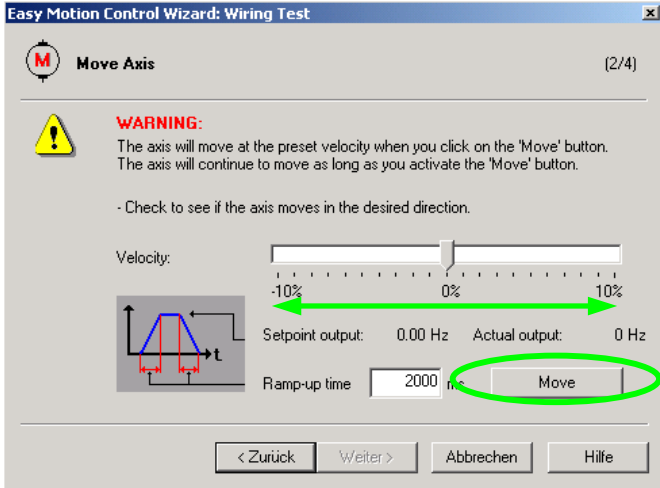
Testen der Verdrahtung / Easy Motion Control Funktionalität

Um die grundsätzliche Funktionalität von Easy Motion Control zu prüfen, empfiehlt sich die Inbetriebnahmefunktion **Verdrahtungstest** der EMC Software. Gehen Sie zum Verdrahtungstest wie folgt vor:

Tabelle 5-1

Nr.	Aktion
1	Verbinden Sie Ihr PG über ein PROFIBUS-Kabel mit dem CP5613 A2 der Station 1.
2	Öffnen Sie, falls noch nicht geschehen, das STEP 7 Projekt „WinLC_Motion“.
3	Richten Sie im STEP 7 Projekt das „PG/PC(1)“ Objekt so ein, dass die PROFIBUS-Schnittstelle aktiv ist.
4	Klicken Sie im SIMATIC Manager auf den Button „online“  . Darauf hin wird ihnen die WinAC RTX im Onlinezustand angezeigt.
5	Löschen Sie nun im Online Projekt aus dem Bausteincontainer den Organisationsbaustein „OB35“. • Klicken Sie dazu auf den Baustein und drücken Sie die Entf Taste (oder rechte Maustaste „Löschen“) • Bestätigen Sie mit „Ja“.
6	Schließen Sie nun das Onlinefenster wieder.
7	Öffnen Sie den Datenbaustein „DB1“ (Axis) mit einem Doppelklick auf den Baustein im Bausteincontainer des SIMATIC Managers.

Positionieren mit WinAC RTX

8	Klicken Sie auf den Reiter „Inbetriebnahme“ und wählen Sie „ Verdrahtungstest “. Folgen Sie dem Assistenten und beachten Sie die Sicherheitshinweise.
9	Mit folgendem Dialogfeld wird der Verdrahtungstest durchgeführt.  • Stellen Sie zunächst den Schieberegler auf 10%. • Klicken Sie dann auf „Fahren“ und halten sie die Maustaste gedrückt.
10	Der Motor sollte sich nun langsam in die im Bild dargestellte Richtung drehen.  Stellen Sie nun den Schieberegler auf –10% und fahren Sie den Motor auf die gleiche Weise wie oben beschrieben. Beobachten Sie nochmals die Drehrichtung des Motors.
11	Klicken Sie anschließend auf „ Weiter > “ und bestätigen Sie die Abfrage bei richtiger Drehrichtung mit „ Ja “.
12	Schließen Sie nun das Easy Motion Control Dialogfeld wieder.
13	Laden Sie nun den Baustein „OB35“ wieder zurück in die Steuerung. Klicken Sie dazu auf den OB35 und anschließend auf den Button  „Laden“.
14	Mit Hilfe dieses kurzen Checks können Sie die grundsätzliche Verdrahtung der Komponenten und die Funktionsweise von EMC überprüfen.



Die hier beschriebene Inbetriebnahmehilfe bewirkt eine Bewegung Ihrer Achse. Eine fehlerhafte Verdrahtung oder Parametrierung hat eine falsche Drehzahl oder Drehrichtung zur Folge. Tragen Sie deshalb dafür Sorge, dass Ihre Anlage durch die Verstellung nicht beschädigt werden kann, z. B. durch Sicherheitsendschalter, die Ihren Antrieb ausschalten.

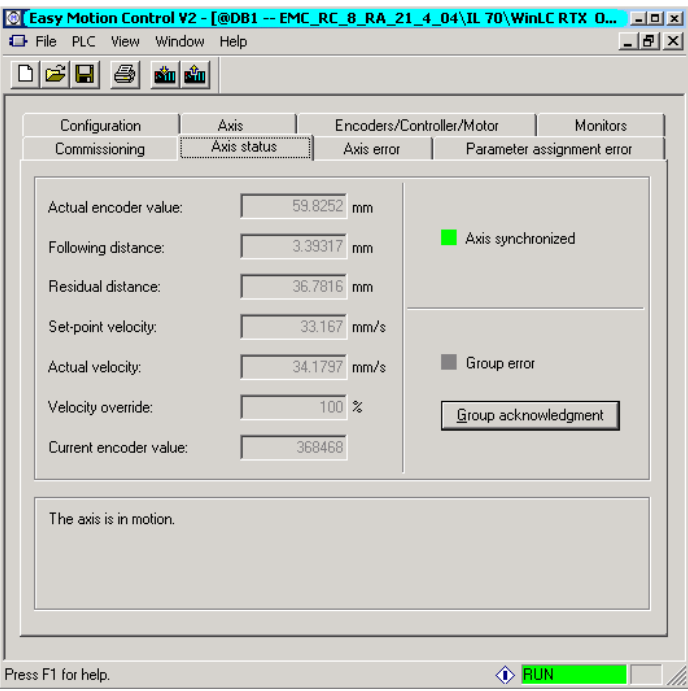
Positionieren mit WinAC RTX

Zustände und Fehlermeldungen der Achse

Ein effektives und umfassendes Beobachten der Achse (Zustände und Achsfehler) ist im Projektierungstool von Easy Motion Control möglich.

Zum Beobachten der Achse gehen Sie wie folgt vor:

Tabelle 5-2

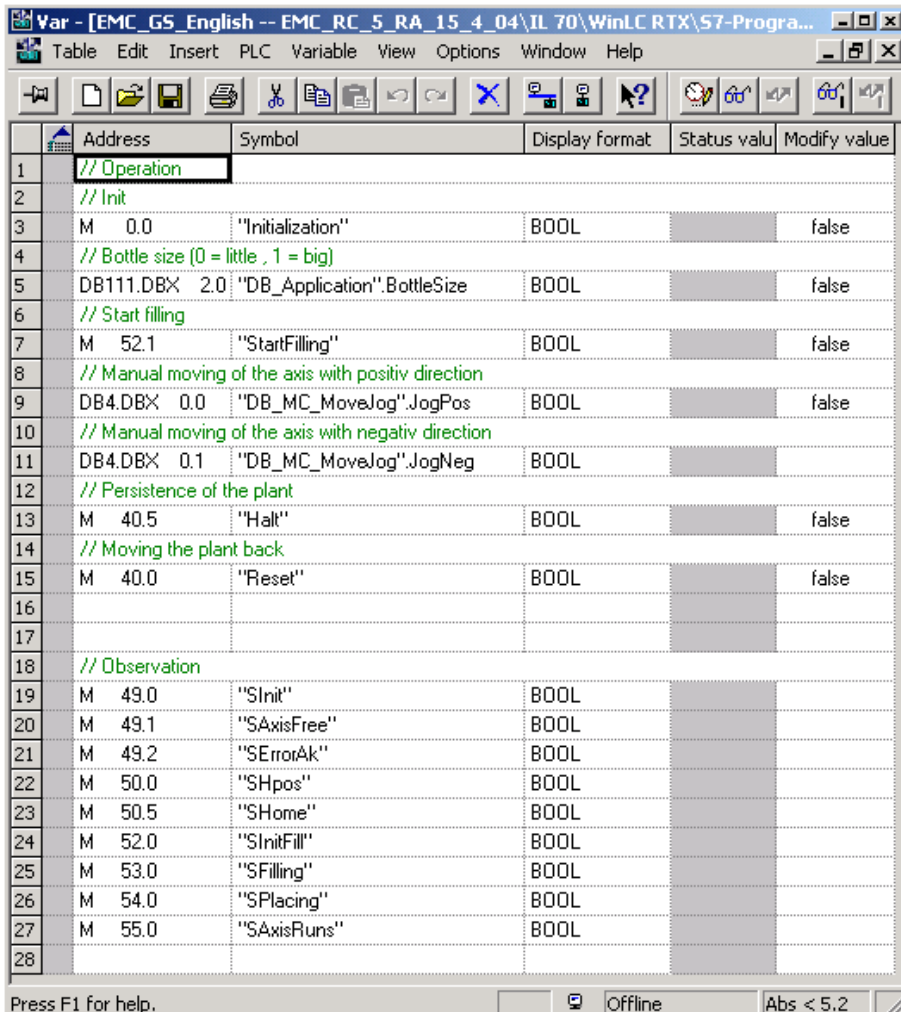
Nr.	Aktion
1	Verbinden Sie Ihr PG über ein PROFIBUS-Kabel mit dem CP5613 A2 der Station 1.
2	Öffnen Sie, falls noch nicht geschehen, das STEP 7 Projekt „WinLC_Motion“.
3	Richten Sie im STEP 7 Projekt das „PG/PC(1)“ Objekt so ein, dass die PROFIBUS-Schnittstelle aktiv ist.
4	Klicken sie im STEP 7 im Bausteincontainer doppelt auf den Baustein „DB1“ Axis .
5	Nach dem Klicken auf den Reiter „Achsstatus“ können die verschiedenen Aktualwerte der Achse beobachtet werden.  Die Bedeutung der einzelnen Aktualwerte entnehmen Sie bitte der Online Hilfe oder dem Easy Motion Control Handbuch.
6	Der Reiter „ Achsfehler “ zeigt die verschiedenen Laufzeitfehler der Achse an.
7	Fehler in der Parametrierung lassen sich über den Reiter „ Parametrierfehler “ ermitteln.

Positionieren mit WinAC RTX

Aktivierung der Variablentabelle

Um die Beispielapplikation mit Hilfe der vorgefertigten Variablentabelle nutzen zu können, führen Sie bitte folgende Schritte aus.

Tabelle 5-3

Nr.	Aktion
1	Verbinden Sie Ihr PG über ein PROFIBUS-Kabel mit dem CP5613 A2 der Station 1.
2	Öffnen Sie, falls noch nicht geschehen, das STEP 7 Projekt „WinLC_Motion“.
3	Richten Sie im STEP 7 Projekt das „PG/PC(1)“ Objekt so ein, dass die PROFIBUS-Schnittstelle aktiv ist.
4	Öffnen Sie die Variablentabelle „EMC_GS_German“ (oder „EMC_GS_English“ für die englische Version) im Bausteinordner der „PCWinAC“ Station.  The screenshot shows the 'Var - [EMC_GS_English -- EMC_RC_5_RA_15_4_04\IL 70\WinLC RTX\57-Progra...' window. It contains a table with the following columns: Address, Symbol, Display format, Status valu, and Modify value. The table lists 28 rows of variables, including initialization, bottle size, start filling, manual moving, persistence of the plant, halt, reset, and observation variables.
5	Starten Sie die Funktion „ Variable beobachten “ über den Button  .

Positionieren mit WinAC RTX

Nr.	Aktion
6	Setzen sie den Merker „Initialization“ auf „1“ und klicken Sie auf den Button „Steuern“ . Hinweis Zum Einfachen Steuern der Werte benutzen Sie die Tastenkombinationen: „<Strg> + 1“ für den Wert „true“ und „<Strg> + 0“ für den Wert „false“
8	Der Schritt „SHpos“ sollte nun aktiv sein. Hinweis Durch die schnelle Verarbeitung in der CPU werden die ersten drei Schritte nur sehr kurz angezeigt. Sie dienen in der Variablen-tabelle lediglich zur Diagnose.
9	Mit Hilfe der beiden Variablen „JogPos“ und „JogNeg“ können Sie nun die Achse von Hand in die gewünschte Ausgangsposition fahren. Setzen Sie für die gewünschte Richtung jeweils das Bit auf „1“ zum Fahren und auf „0“ zum Stoppen. Hinweis Mit Hilfe dieser beiden Funktionen können Sie die grundsätzliche Funktion der Anwendung sowie die Verdrahtung des Aufbaus testen. Setzen Sie die Bits aus Sicherheitsgründen zunächst nur kurz, da der Motor bis auf die maximal eingestellte Drehzahl hochfährt. (setzen = „STRG „+ „1“ / zurück-setzen = „STRG“ + „0“)
10	Wenn die gewünschte Position erreicht ist, kann mit der Variable „BottleSi-ze“ die Größe der Flaschen eingestellt werden. (0 = kleine Flaschen, 1 = große Flaschen)
11	Ist die Flaschenart gewählt, kann mit der Variable „StartFilling“ der Abfüll-vorgang gestartet werden. Setzen Sie das Bit anschließend wieder zurück.
12	Die beiden Zustände (Schritte) „SFilling“ und „SPlacing“ wechseln sich nun immer fortlaufend ab. Die Länge der jeweiligen Schritte hängt von der gewählten Flaschengröße ab.
13	Zum Abstoppen der Anlage aktivieren sie das „Halt“ Bit, worauf die Anlage in den Stop Zustand fährt. Die Anlage befindet sich nun wieder im Schritt „SHome“. Sie kann nun wieder von Hand verfahren oder erneut gestartet werden.
14	Durch Aktivieren des „Reset“ Bits kann die Anlage schnell abgeschaltet werden. Sie muss anschließend allerdings wieder initialisiert werden, da der Antrieb und alle Schritte damit zurückgesetzt werden. Setzen Sie das Bit anschließend wieder zurück.

Positionieren mit WinAC RTX

Bedeutung der Parameter in der Variablentabelle

Es werden folgende Parameter angezeigt:

Tabelle 5-4

Parameter	Beschreibung
Initialization	Mit diesem Merkerbit wird die Initialisierung der Easy Motion Control Bausteine gestartet.
Bottlesize	Durch das Parameterbit Bottlesize wird die in der Anlage verwendete Flaschengröße eingestellt. (0 = kleine Flasche, 1 = grosse Flasche).
StartFilling	Mit dem Merker StartFilling wird der Abfüllvorgang gestartet.
JogPos	Mit diesem Datenbausteinbit kann der Motor von Hand an eine Position in positiver Richtung verfahren.
JogNeg	Mit diesem Datenbausteinbit kann der Motor von Hand an eine Position in negativer Richtung verfahren.
Halt	Der Halt Merker wird zur geregelten Fahrt in den Stop verwendet. Verwendung könnte diese Funktion zum Anhalten der Anlage bei Flaschenwechsel ohne neue Initialisierung finden.
Reset	Mit Reset kann die Anlage zurückgesetzt oder schnell abgestoppt werden. Nach einem Reset muss immer wieder eine Initialisierung vorgenommen werden!
SInit, SAxisFree, SErrorAk, SHpos, SHome, SInitFill, Sfilling, SPlacing, SAxisRuns	Mit diesen Bits können die verschiedenen Schritte der Kette überprüft werden. Eine detaillierte Erklärung kann in Kapitel 3.5 unter Schrittfolge des Anwenders Bausteins FB111 nachgeschlagen werden.
"DB_Application".TTstay1, "DB_Application".pos1, "DB_Application".TTstay2, "DB_Application".pos2	Über diese Parameter können Sie die Dimensionen der beiden Flaschengrößen anpassen. Mit TTstay1 wird die Abfüllzeit der kleinen Flaschen festgelegt. Der Parameter pos1 legt die zu fahrende, relative Strecke (Abstand zwischen den Flaschen) für die kleinen Flaschen fest. Analog gelten die beiden Parameter mit der Nummer 2 für die großen Flaschen.

Hinweis

Durch Anwendung einer geberlosen Vektorregelung kann der Motor während des SHpos Schrittes leicht in seiner Position schwanken. Für diesen Fall führen Sie bitte eine Regleroptimierung (siehe nächster Abschnitt) durch.

6 Literaturangaben

Diese Liste ist keinesfalls vollständig und spiegelt nur eine Auswahl an geeigneter Literatur wieder. Die meisten Handbücher finden Sie, wenn Sie das entsprechende Produkt installiert haben, unter:

Start → Simatic → Dokumentation → [Sprache]

Den Produktsupport finden Sie im Internet unter:

<http://support.automation.siemens.com> (Geben Sie dort die Beitrags ID in das Suchfeld ein).

Tabelle 6-1

Nr.	Themengebiet	Titel
1	Beschreibung der Funktionen und der Bedienung von WinAC RTX V4.0. Zu finden auf der WinAC RTX V4.0 CD.	SIMATIC WinAC RTX V4.0
2	Beschreibung bzw. Informationen zu: - Allgemeine Informationen über die PC-Werkzeuge - Funktionen von NCM PC Im Produktsupport unter der Beitrags ID: 13542666 zu finden.	SIMATIC NET PC-Stationen in Betrieb nehmen - Anleitung und Schnelleinstieg für SIMATIC NCM PC / STEP 7 ab Version V5.2
3	Überblick über die Kommunikation mit SIMATIC S7 / M7 / C7. Im Produktsupport unter der Beitrags ID: 1254686 zu finden.	Kommunikation mit SIMATIC
4	Vollständiger Überblick über die in den Betriebssystemen der CPUs der S7-300 und S7-400 enthaltenen Organisationsbausteine (OB), Systemfunktionen (SFC), System- und Standardfunktionsbausteine (SFB) sowie IEC-Funktionen. Im Produktsupport unter der Beitrags ID: 1214574 zu finden.	Systemsoftware für S7-300/400 System- und Standardfunktionen
5	Handbuch für SIMATIC Rack PC IL40S Im Produktsupport unter der Beitrags ID: 15317654 zu finden.	SIMATIC Rack PC IL40S Handbuch
6	Installationsanleitung für den CP5613 A2 Im Produktsupport unter der Beitrags ID: 13664901 zu finden.	SIMATIC NET Produktinformation / Installationsanleitung für CP5613, CP5614, CP5613 FO, CP5614 FO
7	Handbuch ET 200S Im Produktsupport unter der Beitrags ID: 1144348 zu finden.	Betriebsanleitung: Dezentrales Peripheriesystem ET 200S
8	MICROMASTER 440 Im Produktsupport unter der Beitrags ID: 17142454 zu finden.	Betriebsanleitung MICROMASTER 440; 0,12 KW – 250 kW
9	PROFIBUS- Anschaltung Im Produktsupport unter der Beitrags ID: 6586565 zu finden.	Betriebsanleitung MICROMASTER PROFIBUS-Optionsbaugruppe